



A11103 386421

REFERENCE

NIST
PUBLICATIONS

NBSIR 84-2824

Guide to Available Mathematical Software

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Applied Mathematics
Scientific Computing Division
Washington, DC 20234

Issued January 1984



U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

QC

100

.U56

84-2824

1984

NATIONAL INSTITUTE OF STANDARDS &
TECHNOLOGY

Research Information Center
Gaithersburg, MD 20899

101-51
0-203
0-203
84-2824
1734

NBSIR 84-2824

GUIDE TO AVAILABLE MATHEMATICAL SOFTWARE

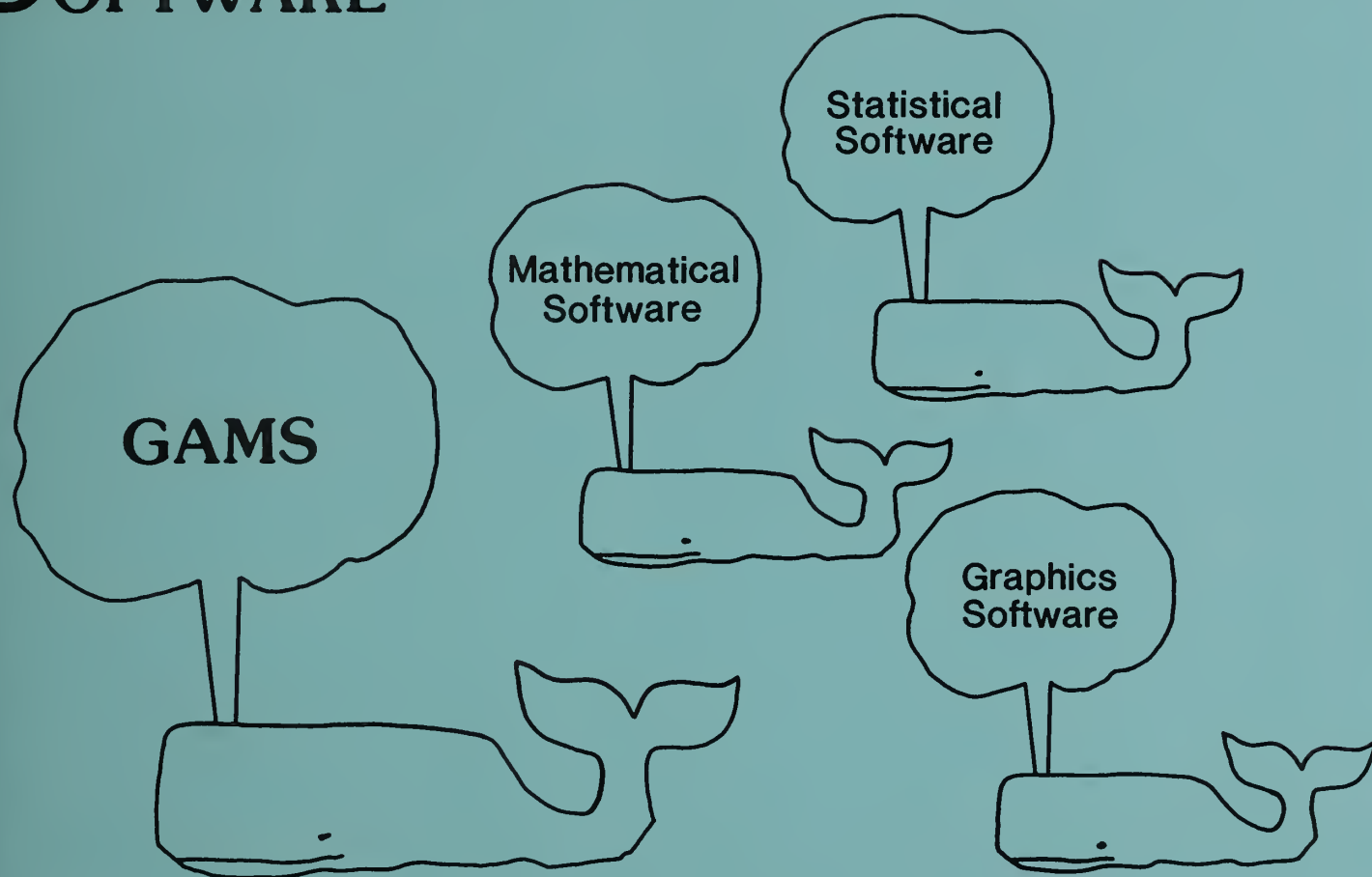
Ronald F. Boisvert
Sally E. Howe
David K. Kahaner

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Applied Mathematics
Scientific Computing Division
Washington, DC 20234

Issued January 1984

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

GUIDE TO AVAILABLE MATHEMATICAL SOFTWARE



U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Center for Applied Mathematics
Washington, D.C. 20234

The Guide to Available Mathematical Software (GAMS) documents software that has been made available for use by the staff of the National Bureau of Standards by its Center for Applied Mathematics. Certain software products, both public-domain and commercial, are identified in GAMS in order to adequately document this software. Identification of such products does not imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that the identified software is necessarily the best available for staff purposes. Conversely, the omission of any such software does not imply its unsuitability for use. The GAMS staff welcome information about scientific software of possible use to NBS staff.

PREFACE

Recent advances in the mathematical sciences have resulted in the emergence of a vast body of reliable and well-designed computer software for aiding in the solution of scientific problems. This software represents both a substantial investment in time and money for its development and maintenance, and a substantial potential savings in time and money to users. The second edition of the Guide to Available Mathematical Software (GAMS) is part of an on-going effort at the National Bureau of Standards to catalog general-purpose mathematical and statistical software used by NBS staff at both the Gaithersburg and the Boulder sites.

The first edition of GAMS contained information about five subprogram libraries: CMLIB, IMSL, MATHWARE, NAG, and PORT. Documentation for the IMSL, NAG, and PORT libraries has been updated for the second edition to reflect their current versions. Eleven new sublibraries have been added to CMLIB which, like MATHWARE, is a repository for public-domain software. Information about software from ten additional libraries has been added to this edition; these libraries are BMDP, DATAPAC, INVAR, MATLAB, Minitab, PDELIB, PLOD, SLDGL, Spectrlan, and STATLIB. These collections consist not only of subprogram libraries, but also collections of stand-alone programs and interactive systems. In all, 2767 subprograms, programs, and interactive system commands are cataloged here. The inclusion of so much more software prompted a major revision to the classification scheme used to catalog the software.

As with the first edition, this guide to computer software was produced using computer software. "Modules by Class" and the "Module Dictionary" are the output of computer programs that systematically query a database designed for this project using BCS-RIM, a portable proprietary relational information management system. The document was typeset using the \TeX text formatting system.

We appreciate the assistance many people provided in the preparation of this edition of GAMS. We first thank the following NBS staff members for their contributions: Janet Blanchard, James Blue, Paul Boggs, Elsie Clark, Amy Del Giorno, George Dines, Janet Donaldson, Jim Filliben, Karla Hoffman, Ric Jackson, Janice Knapp-Cordes, Martin Knapp-Cordes, John Koontz, Daniel Lozier, Colin MacDonald, Jay Murphy, Dianne O'Leary, George Orwell, Hans Oser, Susan Parker, Bert Rust, John Smith, Irene Stegun, Pete Stewart, Selden Stewart, Linda Sung, Roland Sweet, and Anne Trevey. John Barkley, Richard Mattis, Theresa Rodriguez, and Charles Wilson of the NBS Semiconductor Devices and Circuits Division provided VAX computer facilities. We thank Burton Colvin, Director of the Center for Applied Mathematics (CAM); Glenn Ingram, Associate Director for Computing, CAM; and Francis Sullivan, Chief of the Scientific Computing Division, CAM; for their support of research in mathematical and statistical software. The following software library developers provided information about their libraries: Brian Ford at NAG, Phyllis Fox at Bell Laboratories, Jim Gentle at IMSL, MaryAnn Hill at BMDP, and Barbara Ryan at the Minitab Project. The Committee on Statistical Algorithms, Statistical Computing Section, American Statistical Association, assisted with the statistics component of this work; Committee members are: John Aleong, Ken Berk, J. Philip Miller, John Monahan, Bill Sallas, and Del Scott. Finally, we thank the members of the SLATEC (Sandia-Los Alamos-Air Force Weapons Laboratory Technical Exchange Committee) Common Math Library Subcommittee. This common Department of Energy Fortran subprogram library forms the major component of the NBS Core Math Library (CMLIB).

As software for solving scientific problems evolves, so will software documentation such as GAMS. A number of high-quality software products used by NBS staff are not included in this edition of GAMS; we hope to add them to future editions. The format may continue to evolve as we learn how to make GAMS more useful and as technology provides new tools. We welcome your suggestions for improvements.

Ronald F. Boisvert
Sally E. Howe
David K. Kahaner

Scientific Computing Division
Center for Applied Mathematics
National Bureau of Standards
U.S. Department of Commerce
Washington, D.C. 20234

January 1984

CONTENTS

GAMS HELP Card	Inside Front Cover
Preface	iii
How to Use GAMS	v
GAMS Classes	A1 - A15
Modules by Class	B0 - B190
Legend	B0
A. Arithmetic, Error Analysis	B1
B. Number Theory	B4
C. Elementary and Special Functions	B5
D. Linear Algebra	B18
E. Interpolation	B54
F. Solution of Nonlinear Equations	B61
G. Optimization	B66
H. Differentiation and Integration	B75
I. Differential and Integral Equations	B85
J. Integral Transforms	B98
K. Approximation	B101
L. Statistics, Probability	B110
M. Simulation, Stochastic Modeling	B173
N. Data Handling	B174
O. Symbolic Computation	B180
P. Computational Geometry	B181
Q. Graphics	B183
R. Service Routines	B188
S. Software Development Tools	B190
Module Dictionary	C0 - C184
Legend	C0
Library Reference	D0 - D21
Legend	D0
Index	E1 - E19

HOW TO USE GAMS

This document is organized so that an NBS staff member can find software that solves a given mathematical or statistical problem. Briefly, given such a problem, a person should

1. find the classification which most clearly identifies the problem—in GAMS CLASSES or the INDEX,
2. find the list of software that solves problems in that classification—in MODULES BY CLASS, and
3. find further information about that software, such as how to obtain detailed documentation or how to access the software—in the MODULE DICTIONARY and the LIBRARY REFERENCE.

The software cataloged in GAMS is organized into fifteen *libraries*. A library can be either a collection of subprograms, a collection of programs, or an interactive system. Brief descriptions of these libraries are given in the LIBRARY REFERENCE section. Each of these libraries is available for use at the Gaithersburg NBS site, and many are also available at the Boulder site.

Each library is a collection of *modules*. Depending upon the type of library, a module can be a single subprogram, a stand-alone program, or a command in an interactive system. In some cases, libraries are partitioned into *sublibraries*. The modules in a given sublibrary come from a single developer and usually solve a narrow range of problems.

The following describes the five sections of GAMS.

A. GAMS CLASSES

This section contains the classification scheme used to organize the software by mathematical problem. The scheme is tree-structured, with the highest level identified by a single letter and corresponding to very general problems such as J (Integral Transforms). Subclasses are identified by alternating letter-and-number combinations; thus, for example, the subclasses of J1 (Fast Fourier transforms) are labeled J1a, J1b, etc. The longer the class identifier, i.e. the lower the level of the tree, the more detailed the problem specification.

B. MODULES BY CLASS

Information about the software cataloged in each class can be obtained from this section, which is organized in the same way as the GAMS classification scheme. For each class, there may be a discussion of the software in the class and/or a list of software.

The list of software for a given class is organized in alphabetical order by library and module name. The software modules in the class are listed along with brief descriptions of what each module does. These lists generally appear at lower levels of the classification scheme, although software modules which solve broad classes of problems may appear listed at higher levels of the classification scheme. Some classes have no modules.

Discussions are included at the highest levels of the classification scheme (major classes A through S have discussion sections, for example). A discussion addresses the computational issues associated with software in its class and subclasses. The discussions usually include literature references. Most of the references are available at the NBS-Gaithersburg Library (Administration Building, Room E120) and at the Department of Commerce Boulder Laboratories Main Library (Radio Building, Room 1202).

C. MODULE DICTIONARY

Information about individual modules can be obtained from this section, which is organized alphabetically by module name. For each module this information includes a brief description, the library (and sublibrary, if appropriate) to which it belongs, portability information, how to obtain detailed documentation, and how to access the software on the Sperry 1100 at NBS. This information is fully described in the MODULE DICTIONARY Legend (page C0).

D. LIBRARY REFERENCE

This section provides information about the libraries to which modules belong. For each library this information includes type (e.g., subprogram library, program library, or interactive system), version, a description and brief listing of the contents, portability information, library developer, and a reference. For each type of computer at NBS on which the library is supported, information is provided about level of NBS support, how to obtain detailed library documentation, and how to gain access to the library on that computer. This information is fully described in the LIBRARY REFERENCE Legend (page D0).

E. INDEX

This section is a key word and phrase index to the classification scheme. It provides an alternative to GAMS CLASSES as a means of finding a particular problem in the classification scheme.

ON-LINE GAMS

An on-line version of GAMS is available to users of the Sperry 1100 at NBS. It is updated regularly and therefore provides the most current information about available software. This interactive program is executed with the command

`@XQT NBS*XLIB$.GAMS` (in EXEC)

or

`CALL GAMS` (in CTS).

Complete usage instructions can be obtained by typing "?" after the program is entered.

A. ARITHMETIC, ERROR ANALYSIS

- A1. Integer
- A2. Rational
- A3. Real
 - A3a. Single precision
 - A3b. Double precision
 - A3c. Extended precision
 - A3d. Extended range
- A4. Complex
 - A4a. Single precision
 - A4b. Double precision
 - A4c. Extended precision
 - A4d. Extended range
- A5. Interval
 - A5a. Real
 - A5b. Complex
- A6. Change of representation
 - A6a. Type conversion
 - A6b. Base conversion
 - A6c. Decomposition, construction
- A7. Sequences (e.g., convergence acceleration)

B. NUMBER THEORY

C. ELEMENTARY AND SPECIAL FUNCTIONS (*search also class L5*)

- C1. Integer-valued functions (e.g., floor, ceiling, factorial, binomial coefficient)
- C2. Powers, roots, reciprocals
- C3. Polynomials
 - C3a. Orthogonal
 - C3a1. Trigonometric
 - C3a2. Chebyshev, Legendre
 - C3a3. Laguerre
 - C3a4. Hermite
 - C3b. Non-orthogonal
- C4. Elementary transcendental functions
 - C4a. Trigonometric, inverse trigonometric
 - C4b. Exponential, logarithmic
 - C4c. Hyperbolic, inverse hyperbolic
 - C4d. Integrals of elementary transcendental functions
- C5. Exponential and logarithmic integrals
- C6. Cosine and sine integrals
- C7. Gamma
 - C7a. Gamma, log gamma, reciprocal gamma
 - C7b. Beta, log beta
 - C7c. Psi function
 - C7d. Polygamma function
 - C7e. Incomplete gamma
 - C7f. Incomplete beta

- C7g. Riemann zeta
- C8. Error functions
 - C8a. Error functions, their inverses, integrals, including the normal distribution function
 - C8b. Fresnel integrals
 - C8c. Dawson's integral
- C9. Legendre functions
- C10. Bessel functions
 - C10a. $J, Y, H^{(1)}, H^{(2)}$
 - C10a1. Real argument, integer order
 - C10a2. Complex argument, integer order
 - C10a3. Real argument, real order
 - C10a4. Complex argument, real order
 - C10a5. Complex argument, complex order
 - C10b. I, K
 - C10b1. Real argument, integer order
 - C10b2. Complex argument, integer order
 - C10b3. Real argument, real order
 - C10b4. Complex argument, real order
 - C10b5. Complex argument, complex order
 - C10c. Kelvin functions
 - C10d. Airy and Scorer functions
 - C10e. Struve, Anger, and Weber functions
 - C10f. Integrals of Bessel functions
- C11. Confluent hypergeometric functions
- C12. Coulomb wave functions
- C13. Jacobian elliptic functions, theta functions
- C14. Elliptic integrals
- C15. Weierstrass elliptic functions
- C16. Parabolic cylinder functions
- C17. Mathieu functions
- C18. Spheroidal wave functions
- C19. Other special functions

D. LINEAR ALGEBRA

- D1. Elementary vector and matrix operations
 - D1a. Elementary vector operations
 - D1a1. Set to constant
 - D1a2. Minimum and maximum components
 - D1a3. Norm
 - D1a3a. L_1 (sum of magnitudes)
 - D1a3b. L_2 (Euclidean norm)
 - D1a3c. L_∞ (maximum magnitude)
 - D1a4. Dot product (inner product)
 - D1a5. Copy or exchange (swap)
 - D1a6. Multiplication by scalar
 - D1a7. Triad ($ax+y$ for vectors x, y and scalar a)
 - D1a8. Elementary rotation (Givens transformation)
 - D1a9. Elementary reflection (Householder transformation)
 - D1a10. Convolutions
 - D1b. Elementary matrix operations

- D1b1. Set to zero, to identity
- D1b2. Norm
- D1b3. Transpose
- D1b4. Multiplication by vector
- D1b5. Addition, subtraction
- D1b6. Multiplication
- D1b7. Matrix polynomial
- D1b8. Copy
- D1b9. Storage mode conversion
- D1b10. Elementary rotation (Givens transformation)
- D1b11. Elementary reflection (Householder transformation)
- D2. Solution of systems of linear equations (including inversion, LU and related decompositions)
 - D2a. Real nonsymmetric matrices
 - D2a1. General
 - D2a2. Banded
 - D2a2a. Tridiagonal
 - D2a3. Triangular
 - D2a4. Sparse
 - D2b. Real symmetric matrices
 - D2b1. General
 - D2b1a. Indefinite
 - D2b1b. Positive definite
 - D2b2. Positive definite banded
 - D2b2a. Tridiagonal
 - D2b4. Sparse
 - D2c. Complex non-Hermitian matrices
 - D2c1. General
 - D2c2. Banded
 - D2c2a. Tridiagonal
 - D2c3. Triangular
 - D2c4. Sparse
 - D2d. Complex Hermitian matrices
 - D2d1. General
 - D2d1a. Indefinite
 - D2d1b. Positive definite
 - D2d2. Positive definite banded
 - D2d2a. Tridiagonal
 - D2d4. Sparse
 - D2e. Associated operations (e.g., matrix reorderings)
- D3. Determinants
 - D3a. Real nonsymmetric matrices
 - D3a1. General
 - D3a2. Banded
 - D3a2a. Tridiagonal
 - D3a3. Triangular
 - D3a4. Sparse
 - D3b. Real symmetric matrices
 - D3b1. General
 - D3b1a. Indefinite
 - D3b1b. Positive definite
 - D3b2. Positive definite banded
 - D3b2a. Tridiagonal
 - D3b4. Sparse
 - D3c. Complex non-Hermitian matrices
 - D3c1. General
 - D3c2. Banded

- D3c2a. Tridiagonal
- D3c3. Triangular
- D3c4. Sparse
- D3d. Complex Hermitian matrices
- D3d1. General
- D3d1a. Indefinite
- D3d1b. Positive definite
- D3d2. Positive definite banded
- D3d2a. Tridiagonal
- D3d4. Sparse
- D4. Eigenvalues, eigenvectors
- D4a. Ordinary eigenvalue problems ($Ax = \lambda x$)
- D4a1. Real symmetric
- D4a2. Real nonsymmetric
- D4a3. Complex Hermitian
- D4a4. Complex non-Hermitian
- D4a5. Tridiagonal
- D4a6. Banded
- D4a7. Sparse
- D4b. Generalized eigenvalue problems (e.g., $Ax = \lambda Bx$)
- D4b1. Real symmetric
- D4b2. Real general
- D4b3. Complex Hermitian
- D4b4. Complex general
- D4b5. Banded
- D4c. Associated operations
- D4c1. Transform problem
- D4c1a. Balance matrix
- D4c1b. Reduce to compact form
- D4c1b1. Tridiagonal
- D4c1b2. Hessenberg
- D4c1b3. Other
- D4c1c. Standardize problem
- D4c2. Compute eigenvalues of matrix in compact form
- D4c2a. Tridiagonal
- D4c2b. Hessenberg
- D4c2c. Other
- D4c3. Form eigenvectors from eigenvalues
- D4c4. Back transform eigenvectors
- D4c5. Determine Jordan normal form
- D5. QR decomposition, Gram-Schmidt orthogonalization
- D6. Singular value decomposition
- D7. Update matrix decompositions
- D7a. LU
- D7b. Cholesky
- D7c. QR
- D7d. Singular value
- D8. Other matrix equations (e.g., $AX + XB = C$)
- D9. Overdetermined or underdetermined systems of equations, singular systems, pseudo-inverses (*search also classes D5, D6, D9, K1a, L8a*)

E. INTERPOLATION

- E1. Univariate data (curve fitting)
 - E1a. Polynomial splines (piecewise polynomials)
 - E1b. Polynomials
 - E1c. Other functions (e.g., rational, trigonometric)
- E2. Multivariate data (surface fitting)
 - E2a. Gridded
 - E2b. Scattered
- E3. Service routines (e.g., grid generation, evaluation of fitted functions) (*search also class N5*)

F. SOLUTION OF NONLINEAR EQUATIONS

- F1. Single equation
 - F1a. Smooth
 - F1a1. Polynomial
 - F1a1a. Real coefficients
 - F1a1b. Complex coefficients
 - F1a2. Nonpolynomial
 - F1b. General (no smoothness assumed)
- F2. System of equations
 - F2a. Smooth
 - F2b. General (no smoothness assumed)
- F3. Service routines (e.g., check user-supplied derivatives)

G. OPTIMIZATION (*search also classes K, L8*)

- G1. Unconstrained
 - G1a. Univariate
 - G1a1. Smooth function
 - G1a1a. User provides no derivatives
 - G1a1b. User provides first derivatives
 - G1a1c. User provides first and second derivatives
 - G1a2. General function (no smoothness assumed)
 - G1b. Multivariate
 - G1b1. Smooth function
 - G1b1a. User provides no derivatives
 - G1b1b. User provides first derivatives
 - G1b1c. User provides first and second derivatives
 - G1b2. General function (no smoothness assumed)
- G2. Constrained
 - G2a. Linear programming
 - G2a1. Dense matrix of constraints
 - G2a2. Sparse matrix of constraints
 - G2b. Transportation and assignments problem
 - G2c. Integer programming
 - G2c1. Zero/one
 - G2c2. Covering and packing problems
 - G2c3. Knapsack problems
 - G2c4. Matching problems
 - G2c5. Routing, scheduling, location problems
 - G2c6. Pure integer programming
 - G2c7. Mixed integer programming
 - G2d. Network (*for network reliability search class M*)
 - G2d1. Shortest path

- G2d2. Minimum spanning tree
- G2d3. Maximum flow
- G2d3a. Generalized networks
- G2d3b. Networks with side constraints
- G2d4. Test problem generation
- G2e. Quadratic programming
 - G2e1. Positive definite Hessian (i.e. convex problem)
 - G2e2. Indefinite Hessian
- G2f. Geometric programming
- G2g. Dynamic programming
- G2h. General nonlinear programming
- G2h1. Simple bounds
 - G2h1a. Smooth function
 - G2h1a1. User provides no derivatives
 - G2h1a2. User provides first derivatives
 - G2h1a3. User provides first and second derivatives
 - G2h1b. General function (no smoothness assumed)
- G2h2. Linear equality or inequality constraints
 - G2h2a. Smooth function
 - G2h2a1. User provides no derivatives
 - G2h2a2. User provides first derivatives
 - G2h2a3. User provides first and second derivatives
 - G2h2b. General function (no smoothness assumed)
- G2h3. Nonlinear constraints
 - G2h3a. Equality constraints only
 - G2h3a1. Smooth function and constraints
 - G2h3a1a. User provides no derivatives
 - G2h3a1b. User provides first derivatives of function and constraints
 - G2h3a1c. User provides first and second derivatives of function and constraints
 - G2h3a2. General function and constraints (no smoothness assumed)
 - G2h3b. Equality and inequality constraints
 - G2h3b1. Smooth function and constraints
 - G2h3b1a. User provides no derivatives
 - G2h3b1b. User provides first derivatives of function and constraints
 - G2h3b1c. User provides first and second derivatives of function and constraints
 - G2h3b2. General function and constraints (no smoothness assumed)
- G2i. Global solution to nonconvex problems
- G3. Optimal control
- G4. Service routines
 - G4a. Problem input (e.g., matrix generation)
 - G4b. Problem scaling
 - G4c. Check user-supplied derivatives
 - G4d. Find feasible point
 - G4e. Check for redundancy
 - G4f. Other

H. DIFFERENTIATION, INTEGRATION

- H1. Numerical differentiation
- H2. Quadrature (numerical evaluation of definite integrals)
 - H2a. One-dimensional integrals
 - H2a1. Finite interval (general integrand)
 - H2a1a. Integrand available via user-defined procedure
 - H2a1a1. Automatic (user need only specify required accuracy)
 - H2a1a2. Nonautomatic

- H2a1b. Integrand available only on grid
- H2a1b1. Automatic (user need only specify required accuracy)
- H2a1b2. Nonautomatic
- H2a2. Finite interval (specific or special type integrand including weight functions, oscillating and singular integrands, principal value integrals, splines, etc.)
- H2a2a. Integrand available via user-defined procedure
- H2a2a1. Automatic (user need only specify required accuracy)
- H2a2a2. Nonautomatic
- H2a2b. Integrand available only on grid
- H2a2b1. Automatic (user need only specify required accuracy)
- H2a2b2. Nonautomatic
- H2a3. Semi-infinite interval (including e^{-x} weight function)
- H2a3a. Integrand available via user-defined procedure
- H2a3a1. Automatic (user need only specify required accuracy)
- H2a3a2. Nonautomatic
- H2a4. Infinite interval (including e^{-x^2} weight function)
- H2a4a. Integrand available via user-defined procedure
- H2a4a1. Automatic (user need only specify required accuracy)
- H2a4a2. Nonautomatic
- H2b. Multidimensional integrals
- H2b1. One or more hyper-rectangular regions
- H2b1a. Integrand available via user-defined procedure
- H2b1a1. Automatic (user need only specify required accuracy)
- H2b1a2. Nonautomatic
- H2b1b. Integrand available only on grid
- H2b1b1. Automatic (user need only specify required accuracy)
- H2b1b2. Nonautomatic
- H2b2. Nonrectangular region, general region
- H2b2a. Integrand available via user-defined procedure
- H2b2a1. Automatic (user need only specify required accuracy)
- H2b2a2. Nonautomatic
- H2b2b. Integrand available only on grid
- H2b2b1. Automatic (user need only specify required accuracy)
- H2b2b2. Nonautomatic
- H2c. Service routines (compute weight and nodes for quadrature formulas)

I. DIFFERENTIAL AND INTEGRAL EQUATIONS

- I1. Ordinary differential equations
 - I1a. Initial value problems
 - I1a1. General, nonstiff or mildly stiff
 - I1a1a. One-step methods (e.g., Runge-Kutta)
 - I1a1b. Multistep methods (e.g., Adams' predictor-corrector)
 - I1a1c. Extrapolation methods (e.g., Bulirsch-Stoer)
 - I1a2. Stiff and mixed algebraic-differential equations
 - I1b. Multipoint boundary value problems
 - I1b1. Linear
 - I1b2. Nonlinear
 - I1b3. Eigenvalue (e.g., Sturm-Liouville)
 - I1c. Service routines (e.g., interpolation of solutions, error handling)
- I2. Partial differential equations
 - I2a. Initial boundary value problems
 - I2a1. Parabolic
 - I2a1a. One spatial dimension
 - I2a1b. Two or more spatial dimensions
 - I2a2. Hyperbolic

- I2b. Elliptic boundary value problems
 - I2b1. Linear
 - I2b1a. Second order
 - I2b1a1. Poisson (Laplace) or Helmholtz equation
 - I2b1a1a. Rectangular domain (or topologically rectangular in the coordinate system)
 - I2b1a1b. Nonrectangular domain
 - I2b1a2. Other separable problems
 - I2b1a3. Nonseparable problems
 - I2b1c. Higher order equations (e.g., biharmonic)
 - I2b2. Nonlinear
 - I2b3. Eigenvalue
 - I2b4. Service routines
 - I2b4a. Domain triangulation (*search also class P2a2c1*)
 - I2b4b. Solution of discretized elliptic equations
- I3. Integral equations

J. INTEGRAL TRANSFORMS

- J1. Fast Fourier transforms (*search class L10 for time series analysis*)
 - J1a. One-dimensional
 - J1a1. Real
 - J1a2. Complex
 - J1a3. Trigonometric (sine, cosine)
 - J1b. Multidimensional
- J2. Convolutions
- J3. Laplace transforms
- J4. Hilbert transforms

K. APPROXIMATION (*search also class L8*)

- K1. Least squares (L_2) approximation
 - K1a. Linear least squares (*search also classes D5, D6, D9*)
 - K1a1. Unconstrained
 - K1a1a. Univariate data (curve fitting)
 - K1a1a1. Polynomial splines (piecewise polynomials)
 - K1a1a2. Polynomials
 - K1a1a3. Other functions (e.g., rational, trigonometric, user-specified)
 - K1a1b. Multivariate data (surface fitting)
 - K1a2. Constrained
 - K1a2a. Linear constraints
 - K1a2b. Nonlinear constraints
 - K1b. Nonlinear least squares
 - K1b1. Unconstrained
 - K1b1a. Smooth functions
 - K1b1a1. User provides no derivatives
 - K1b1a2. User provides first derivatives
 - K1b1a3. User provides first and second derivatives
 - K1b1b. General functions
 - K1b2. Constrained
 - K1b2a. Linear constraints
 - K1b2b. Nonlinear constraints
- K2. Minimax (L_∞) approximation
- K3. Least absolute value (L_1) approximation

- K4. Other analytic approximations (e.g., Taylor polynomial, Pade)
- K5. Smoothing
- K6. Service routines (e.g., mesh generation, evaluation of fitted functions) (*search also class N5*)

L. STATISTICS, PROBABILITY

- L1. Data summarization
 - L1a. One univariate quantitative sample
 - L1a1. Ungrouped data
 - L1a1a. Location
 - L1a1b. Dispersion
 - L1a1c. Shape
 - L1a1d. Distribution, density
 - L1a2. Ungrouped data with missing values
 - L1a3. Grouped data
 - L1a3a. Location
 - L1a3b. Dispersion
 - L1a3c. Shape
 - L1c. One univariate qualitative (proportional) sample
 - L1e. Two or more univariate samples or one multivariate sample
 - L1e1. Ungrouped data
 - L1e1a. Location
 - L1e1b. Correlation
 - L1e2. Ungrouped data with missing values
 - L1e3. Grouped data
 - L1f. Two or more multivariate samples
- L2. Data manipulation (*search also class N*)
 - L2a. Transform (*search also class N6 for sorting, ranking*)
 - L2b. Group
 - L2c. Sample
 - L2d. Subset
- L3. Graphics (*search also class Q*)
 - L3a. Histograms
 - L3b. Distribution functions
 - L3c. Scatter diagrams
 - L3c1. y vs. x
 - L3c2. Symbol plots
 - L3c3. Multiple plots
 - L3c4. Probability plots
 - L3c4b. Beta, binomial
 - L3c4c. Cauchy, chi-squared
 - L3c4d. Double exponential
 - L3c4e. Exponential, extreme value
 - L3c4f. F distribution
 - L3c4g. Gamma, geometric
 - L3c4h. Halfnormal
 - L3c4l. Lambda, logistic, lognormal
 - L3c4n. Negative binomial, normal
 - L3c4p. Pareto, Poisson
 - L3c4t. t distribution
 - L3c4u. Uniform
 - L3c4w. Weibull
 - L3c5. Time series plots (x_i vs. i , vertical, lag)
 - L3d. EDA graphics

- L4. Elementary statistical inference, hypothesis testing
 - L4a. One univariate quantitative sample
 - L4a1. Ungrouped data
 - L4a1a. Parameter estimation
 - L4a1a2. Binomial
 - L4a1a5. Extreme value
 - L4a1a14. Normal
 - L4a1a16. Poisson
 - L4a1a21. Uniform
 - L4a1a23. Weibull
 - L4a1b. Distribution-free (nonparametric) analysis
 - L4a1c. Goodness-of-fit tests
 - L4a1d. Tests on sequences of numbers
 - L4a1e. Density and distribution function estimation
 - L4a1f. Tolerance limits
 - L4a2. Ungrouped data with missing values
 - L4a3. Grouped data
 - L4a3a. Parameter estimation
 - L4a3a14. Normal
 - L4b. Two or more univariate quantitative samples
 - L4b1. Ungrouped data
 - L4b1a. Parameter estimation
 - L4b1a14. Normal
 - L4b1b. Distribution-free (nonparametric) analysis
 - L4b2. Ungrouped data with missing values
 - L4b3. Grouped data
 - L4c. One univariate qualitative (proportional) sample
 - L4d. Two or more univariate samples
 - L4e. One multivariate sample
 - L4e1. Ungrouped data
 - L4e1a. Parameter estimation
 - L4e1a14. Normal
 - L4e1b. Distribution-free (nonparametric) analysis
 - L4e2. Ungrouped data with missing values
 - L4e2a. Parameter estimation
 - L4e2b. Distribution-free (nonparametric) analysis
 - L4e3. Grouped data
 - L4e3a. Parameter estimation
 - L4e3a14. Normal
 - L4e3b. Distribution-free (nonparametric) analysis
 - L4e4. Two or more multivariate samples
 - L4e4a. Parameter estimation
 - L4e4a14. Normal
- L5. Function evaluation (*search also class C*)
 - L5a. Univariate
 - L5a1. Cumulative distribution functions, probability density functions
 - L5a1b. Beta, binomial
 - L5a1c. Cauchy, chi-squared
 - L5a1d. Double exponential
 - L5a1e. Error function, exponential, extreme value
 - L5a1f. F distribution
 - L5a1g. Gamma, general, geometric
 - L5a1h. Halfnormal, hypergeometric
 - L5a1k. Kolmogorov-Smirnov
 - L5a1l. Lambda, logistic, lognormal
 - L5a1n. Negative binomial, normal

L5a1p.	Pareto, Poisson
L5a1t.	t distribution
L5a1u.	Uniform
L5a1w.	Weibull
L5a2.	Inverse cumulative distribution functions, sparsity functions
L5a2b.	Beta, binomial
L5a2c.	Cauchy, chi-squared
L5a2d.	Double exponential
L5a2e.	Exponential, extreme value
L5a2f.	F distribution
L5a2g.	Gamma, general, geometric
L5a2h.	Halfnormal
L5a2l.	Lambda, logistic, lognormal
L5a2n.	Negative binomial, normal, normal scores
L5a2p.	Pareto, Poisson
L5a2t.	t distribution
L5a2u.	Uniform
L5a2w.	Weibull
L5b.	Multivariate
L5b1.	Cumulative distribution functions, probability density functions
L5b1n.	Normal
L6.	Pseudo-random number generation
L6a.	Univariate
L6a2.	Beta, binomial, Boolean
L6a3.	Cauchy, chi-squared
L6a4.	Double exponential
L6a5.	Exponential, extreme value
L6a6.	F distribution
L6a7.	Gamma, general (continuous, discrete) distributions, geometric
L6a8.	Halfnormal, hypergeometric
L6a9.	Integers
L6a12.	Lambda, logical, logistic, lognormal
L6a14.	Negative binomial, normal
L6a15.	Order statistics
L6a16.	Pareto, permutations, Poisson
L6a19.	Samples, stable distribution
L6a20.	t distribution, time series, triangular
L6a21.	Uniform
L6a22.	Von Mises
L6a23.	Weibull
L6b.	Multivariate
L6b3.	Contingency table, correlation matrix
L6b13.	Multinomial
L6b14.	Normal
L6b15.	Orthogonal matrix
L6b21.	Uniform
L6c.	Service routines (e.g., seed)
L7.	Experimental design, including analysis of variance
L7a.	Univariate
L7a1.	One-way analysis of variance
L7a1a.	Parametric analysis
L7a1a1.	Contrasts, multiple comparisons
L7a1a2.	Analysis of variance components
L7a1b.	Distribution-free (nonparametric) analysis
L7a2.	Balanced multiway design
L7a2a.	Complete

L7a2a1.	Parametric analysis
L7a2a1a.	Two-way
L7a2a1b.	Factorial
L7a2a1c.	Nested
L7a2a2.	Distribution-free (nonparametric) analysis
L7a2b.	Incomplete
L7a2b1.	Parametric analysis
L7a2b1a.	Latin square
L7a2b1b.	Lattice designs
L7a2b2.	Distribution-free (nonparametric) analysis
L7a3.	Analysis of covariance
L7a4.	General linear model (unbalanced design)
L7a4a.	Parametric analysis
L7a4b.	Distribution-free (nonparametric) analysis
L7b.	Multivariate
L8.	Regression (<i>search also classes G, K</i>)
L8a.	Linear least squares (L_2) (<i>search also classes D5, D6, D9</i>)
L8a1.	Simple (e.g., $y = \beta_0 + \beta_1 x + \epsilon$)
L8a1a.	Ordinary
L8a1a1.	Unweighted
L8a1a1a.	No missing values
L8a1a1b.	Missing values
L8a1a2.	Weighted
L8a1b.	Through the origin
L8a1c.	Errors in variables
L8a1d.	Calibration (inverse regression)
L8a2.	Polynomial (e.g., $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$)
L8a2a.	Not using orthogonal polynomials
L8a2a1.	Unweighted
L8a2a2.	Weighted
L8a2b.	Using orthogonal polynomials
L8a2b1.	Unweighted
L8a2b2.	Weighted
L8a3.	Piecewise polynomial (i.e. multiphase or spline)
L8a4.	Multiple (e.g., $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$)
L8a4a.	Ordinary
L8a4a1.	Unweighted
L8a4a1a.	No missing values
L8a4a1b.	Missing values
L8a4a1c.	From correlation data
L8a4a1d.	Using principal components
L8a4a1e.	Using preference pairs
L8a4a2.	Weighted
L8a4b.	Errors in variables
L8a4d.	Logistic
L8a5.	Variable selection
L8a6.	Regression design
L8a7.	Several multiple regressions
L8a8.	Multivariate
L8a9.	Diagnostics
L8a10.	Hypothesis testing, inference
L8a10a.	Lack-of-fit tests
L8a10b.	Analysis of residuals
L8a10c.	Inference
L8b.	Biased (ridge)
L8c.	Linear least absolute value (L_1)

- L8d. Linear minimax (L_∞)
- L8e. Robust
- L8f. EDA
- L8g. Nonlinear
 - L8g1. Unweighted
 - L8g1a. Derivatives not supplied
 - L8g1b. Derivatives supplied
 - L8g2. Weighted
 - L8g2a. Derivatives not supplied
 - L8g2b. Derivatives supplied
- L8h. Service routines
- L9. Categorical data analysis
 - L9a. 2-by-2 tables
 - L9b. Two-way tables
 - L9c. Log-linear model
 - L9d. EDA (e.g., median polish)
- L10. Time series analysis (*search also class L9c5 for time series graphics*)
 - L10a. Transformations, transforms (*search also class J1*)
 - L10b. Smoothing, filtering
 - L10c. Autocorrelation analysis
 - L10d. Complex demodulation
 - L10e. ARMA and ARIMA modeling and forecasting
 - L10e1. Model and parameter estimation
 - L10e2. Forecasting
 - L10f. Spectral analysis
 - L10g. Cross-correlation analysis
 - L10g1. Parameter estimation
 - L10g2. Forecasting
- L11. Correlation analysis
- L12. Discriminant analysis
- L13. Factor analysis
 - L13a. Principal components analysis
- L14. Cluster analysis
 - L14a. Unconstrained
 - L14a1. Nested
 - L14a1a. Joining (e.g., single link)
 - L14a1b. Divisive
 - L14a2. Non-nested
 - L14b. Constrained
 - L14b1. One-dimensional
 - L14b2. Two-dimensional
 - L14c. Display
- L15. Life testing, survival analysis
- M. SIMULATION, STOCHASTIC MODELING (*search also classes L6, L10*)
 - M1. Simulation
 - M1a. Discrete
 - M1b. Continuous (Markov models)
 - M2. Queueing
 - M3. Reliability
 - M3a. Quality control

M3b. Electrical network

M4. Project optimization (e.g., PERT)

N. DATA HANDLING (*search also class L2*)

N1. Input, output

N2. Bit manipulation

N3. Character manipulation

N4. Storage management (e.g., stacks, heaps, trees)

N5. Searching

N5a. Extreme value

N5b. Insertion position

N5c. On a key

N6. Sorting

N6a. Internal

N6a1. Passive (i.e. construct pointer array, rank)

N6a1a. Integer

N6a1b. Real

N6a1b1. Single precision

N6a1b2. Double precision

N6a1c. Character

N6a2. Active

N6a2a. Integer

N6a2b. Real

N6a2b1. Single precision

N6a2b2. Double precision

N6a2c. Character

N6b. External

N7. Merging

N8. Permuting

O. SYMBOLIC COMPUTATION

P. COMPUTATIONAL GEOMETRY (*search also classes G, Q*)

P1. One dimension

P2. Two dimensions

P2a. Points, lines

P2a1. Relationships

P2a1a. Closest and farthest points

P2a1b. Intersection

P2a2. Graph construction

P2a2a. Convex hull

P2a2b. Minimum spanning tree

P2a2c. Region partitioning

P2a2c1. Triangulation

P2a2c2. Voronoi diagram

P2b. Polygons (e.g., intersection, hidden line problems)

P2c. Circles

P3. Three dimensions

P3a. Points, lines, planes

P3b. Polytopes

P3c. Spheres

P4. More than three dimensions

Q. GRAPHICS (*search also classes L9, P*)

Q1. Line printer plotting

R. SERVICE ROUTINES

R1. Machine-dependent constants

R2. Error checking (e.g., check monotonicity)

R3. Error handling

R3a. Set criteria for fatal errors

R3b. Set unit number for error messages

R3c. Other utility programs

R4. Documentation retrieval

S. SOFTWARE DEVELOPMENT TOOLS

S1. Program transformation

S2. Static analysis

S3. Dynamic analysis

A: Arithmetic, Error Analysis

Modern scientific computers provide facilities to perform the basic arithmetic operations such as addition, multiplication, and division on both fixed and floating-point numbers. Programming languages make these facilities directly accessible through their implementations of integer and real arithmetic. Other arithmetic systems are also useful for scientific computation, however. Complex arithmetic is the prime example; it is available in major scientific programming languages, and many software libraries provide software to support it. Other important examples are multiple precision integer and real arithmetic, rational arithmetic, and interval arithmetic.

Software implementing non-standard, though elementary, arithmetic operations on a variety of data types are classified in this chapter. For example, the NAG and PORT libraries have collections of Fortran subprograms which implement basic arithmetic operations for both single and double precision complex numbers using only real arithmetic.

Conversion from one data type to another is another elementary operation for which software exists. The PORT library, for example, provides a complete set of Fortran subprograms for performing standard type conversion, including the construction of machine-base numbers given a base-10 mantissa and exponent, and the inverse operation.

January 1984

A :	Arithmetic, error analysis
------------	----------------------------

A1 :	Integer
-------------	---------

IMSL subprogram library

VDCPS Decompose an integer into its prime factors.

A2 :	Rational
-------------	----------

A3 :	Real
-------------	------

A3a :	Single precision
--------------	------------------

A3b :	Double precision
--------------	------------------

A3c :	Extended precision
--------------	--------------------

A3d :	Extended range
--------------	----------------

A4 :	Complex
-------------	---------

A4a :	Single precision
--------------	------------------

CMLIB subprogram library (FNLIB sublibrary)

CARG Argument = θ , in radians of complex number, $z = |z|e^{i\theta}$.

NAG subprogram library

- A02ABE** Modulus of a complex number. Double precision version is A02ABF.
A02ACE Quotient of two complex numbers. Double precision version is A02ACF.
C06GBE Complex conjugate of complex data values, Hermitian sequence. Double precision version is C06GBF.
C06GCE Complex conjugate of complex data values, general sequence. Double precision version is C06GCF.

A4b : Double precision

NAG subprogram library

- A02ABF** Modulus of a complex number.
A02ACF Quotient of two complex numbers.
C06GBF Complex conjugate of complex data values, Hermitian sequence.
C06GCF Complex conjugate of complex data values, general sequence.

PORT subprogram library

- CDADD** Adds two complex double precision numbers. Each is represented by a double precision array of two elements.
CDDIV Divides two double precision complex numbers. Each is represented by a double precision array of two elements.
CDMUL Multiplies two double precision complex numbers. Each is represented by a double precision array of two elements.

A4c : Extended precision

A4d : Extended range

A5 : Interval

A5a : Real

A5b : Complex

A6 : Change of representation

A6a : Type conversion

PORT subprogram library

- CNVBDC** Converts values from one vector to another. Backward loop, double precision into complex.
CNVBDI Converts values from one vector to another. Backward loop, double precision into integer.
CNVBDR Converts values from one vector to another. Backward loop, double precision into real.
CNVBIC Converts values from one vector to another. Backward loop, integer into complex.

CNVBID	Converts values from one vector to another. Backward loop, integer to double precision.
CNVBIR	Converts values from one vector to another. Backward loop, integer into real.
CNVBRC	Converts values from one vector to another. Backward loop, real into complex.
CNVBRD	Converts values from one vector to another. Backward loop, real into double precision.
CNVBRI	Converts values from one vector to another. Backward loop, real into integer.
CNVFDC	Converts values from one vector to another. Forward loop, double precision into complex.
CNVFDI	Converts values from one vector to another. Forward loop, double precision into integer.
CNVFDR	Converts values from one vector to another. Forward loop, double precision into real.
CNVFIC	Converts values from one vector to another. Forward loop, integer into complex.
CNVFID	Converts values from one vector to another. Forward loop, integer to double precision.
CNVFIR	Converts values from one vector to another. Forward loop, integer to real.
CNVFRC	Converts values from one vector to another. Forward loop, real into complex.
CNVFRD	Converts values from one vector to another. Forward loop, real into double precision.
CNVFRI	Converts values from one vector to another. Forward loop, real into integer.

A6b : Base conversion**PORT subprogram library**

VBTD	Converts a mantissa and exponent into a base 10 floating point number. Double precision version is DVBTOD.
-------------	--

A6c : Decomposition, construction**PORT subprogram library**

UMKFL	Decomposes a non-zero floating point number into a mantissa and an exponent. Double precision version is DUMKFL.
VDTOB	Converts a base-10 mantissa and exponent of a floating point number into a machine-base representation. Double precision version is DVDTOB.

A7 : Sequences (e.g., convergence acceleration)**NAG subprogram library**

C06BAE	Performs Shanks' transformation on a given sequence of real values by means of the ϵ -algorithm of Wynn. A (possibly unreliable) estimate of the absolute error is also given. An erratic, but often powerful method for accelerating sequences. Double precision version is C06BAF.
---------------	---

B: Number Theory

Software classified in this chapter performs such number-theoretic calculations as the decomposition of integers into prime factors.

January 1984

B :	Number theory
------------	---------------

IMSL subprogram library

VDCPS Decompose an integer into its prime factors.

C: Elementary and Special Functions

This chapter contains computer programs for selected mathematical functions. The provision of such software is still a wide open field. Some idea of the difficulties involved is given in the remainder of this introduction. The reader who is interested only in what is currently available should refer immediately to the categories below for further specific information. For the special distribution functions of statistics see also class L5.

Some elementary functions are provided by the Fortran compiler. Any compiler that conforms to the ANSI standard should include single and double precision, and in some cases single precision complex, versions of the square root function, the exponential and logarithmic functions, the circular functions and inverses, and the hyperbolic functions. Some compilers provide more. For example, Sperry ASCII Fortran (FTN) includes double precision complex versions of some of these functions, plus the error, complementary error, gamma, and log gamma functions in single and double precision.

But when the same Fortran program is to be run on computers made by different manufacturers, it is not wise to rely on any nonstandard feature of the language. Therefore portable Fortran programs for mathematical functions, other than the ones required by the Fortran standard (ANSI X3.9 - 1978), are of great interest. The major implementational difficulty is how to achieve efficiently the requisite accuracy for different computer wordlengths.

This can be seen by considering very briefly how a special function is computed. It is always computed from some form of approximation. Usually the approximation derives from some type of truncated infinite expansion. In general, the more accurate the approximation, the more expensive the computation. The additional expense can be quite considerable, especially when the approximation involves expansions in series of functions instead of just a power series. Thus an efficient portable function must be self-adaptive in the sense of being able to select the most suitable approximation from a set of possible ones.

There are algorithmic difficulties as well. Rounding errors occur in actually computing an approximation to a function in finite precision. These errors accumulate so that the final computed function value may be much less accurate than expected. Good algorithms minimize the rounding error, but some error always remains. Even for relatively simple functions it can be quite large, especially when the number of arithmetic operations that must take place is large. In any case the algorithm that is used should be stable in the sense that it should not amplify rounding errors too much as the computation progresses.

The accuracy of functions implemented in the compiler is generally higher than the accuracy of functions in a portable library. This is because it is more convenient and less costly in assembly language than in Fortran to perform the computation in extended precision.

Progress has been made in meeting these difficulties in recent years. There now exist fairly widely accepted Fortran calls that return static information about the number of bits in a computer word, the exponent range, etc. for both single and double precision. This information has been used effectively in some cases to solve the algorithm selection problem. Research work continues on developing better algorithms and computer arithmetic, on improving the design of Fortran subroutines for mathematical functions, and on developing standards for programming languages and function libraries. This work will lead to mathematical function software that is easier to use, gives fuller coverage of input ranges of interest, and that has more versatile modes of recovery from faults. It will also lead to a much wider selection of mathematical functions in portable libraries.

January 1984

References

1. Abramowitz, M. and I. A. Stegun (Editors) (1964). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, U.S. Government Printing Office, Washington, D.C.
2. Cody, W. J. (1974). The Construction of Numerical Subroutine Libraries, *SIAM Review*, Vol. 13, No. 1, pp. 36-46.
3. Cody, W. J. (1980). Software Basics for Computational Mathematics, *ACM SIGNUM Newsletter*, Vol. 15, No. 2, pp. 18-29.
4. Cody, W. J. and W. M. Waite (1980). *Software Manual for the Elementary Functions*, Prentice-Hall, Englewood Cliffs, New Jersey.
5. Fike, C. T. (1968). *Computer Evaluation of Mathematical Functions*, Prentice-Hall, Englewood Cliffs, New Jersey.
6. Hart, J. F., E. W. Cheney, C. L. Lawson, et al. (1968). *Computer Approximations*, John Wiley & Sons, New York, New York.

C : Elementary and special functions (search also class L5)

C1 : Integer-valued functions (e.g., floor, ceiling, factorial, binomial coefficient)

CMLIB subprogram library (FNLIB sublibrary)

BINOM Binomial $n! / (m! (n-m)!)$. Double precision version is DBINOM.
FAC Factorial, $= n!$. Double precision version is DFAC.
POCH Pochhammer's symbol $(a)_z = \Gamma(a+z)/\Gamma(a)$. Double precision version is DPOCH.
POCH1 Pochhammer's symbol from first order, $= ((a)_z - 1)/z$. Double precision version is DPOCH1.

PORT subprogram library

CEIL Finds the smallest integer greater than or equal to x . Input and output are real. Double precision version is DCEIL.
FLR Finds the largest integer less than or equal to x . Input and output are real. Double precision version is DFLR.
ICEIL Finds the smallest integer greater than or equal to x . Input is real, output is integer.
IDCEIL Finds the smallest integer greater than or equal to x . Input is double precision, output is integer.
IFLR Finds the largest integer less than or equal to x . Input is real, output is integer. Double precision version is IDFLR.

C2 : Powers, roots, reciprocals

CMLIB subprogram library (FNLIB sublibrary)

CBRT Cube root of real number. Double precision version is DCBRT.
CCBRT Complex cube root of complex argument.
SQRT Square root.

NAG subprogram library

A02AAE Evaluates the square root of a complex number. Double precision version is A02AAF.

C3 : Polynomials

PORT subprogram library

ORTH Evaluates a polynomial expressed as a sum of general orthogonal polynomials. Double precision version is DORTH.

C3a : Orthogonal

C3a1 : Trigonometric

PORT subprogram library

TRIGP Evaluates a trigonometric polynomial with given coefficients. Double precision version is DTRIGP.

C3a2 : Chebyshev, Legendre**CMLIB subprogram library (FCNPAK sublibrary)**

XSLEGF Calculates sequences of values of Legendre functions of the first kind of positive or negative order, as well as Legendre functions of the second kind, and normalized Legendre functions. Double precision version is XDLEGF.

XSNRMP Calculates a sequence of values of the normalized Legendre polynomials for fixed degree and argument and variable order. Double precision version is XDNRMP.

CMLIB subprogram library (FNLIB sublibrary)

CSEVL Evaluates an n term series of Chebyshev polynomials at a given point. Double precision version is DCSEVL.

CMLIB subprogram library (QUADSP sublibrary)

QMOMO Computes integral of k-th degree Tchebycheff polynomial times selection of functions with various singularities. Double precision version is DQMOMO.

IMSL subprogram library

RLPOL Generate orthogonal (Gram) polynomials on a given set of abscissas

NAG subprogram library

C06DBE Sum of a Chebyshev series. Double precision version is C06DBF.

E02AEE Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form (simplified parameter list). Double precision version is E02AEF.

E02AHE Derivative of fitted polynomial in Chebyshev series form. Double precision version is E02AHF.

E02AJE Integral of fitted polynomial in Chebyshev series form. Double precision version is E02AJF.

E02AKE Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form. Double precision version is E02AKF.

PORT subprogram library

TCHBP Evaluates a polynomial expressed as a sum of Chebyshev polynomials. Double precision version is DTCHBP.

C3a3 : Laguerre**C3a4 :** Hermite**C3b :** Non-orthogonal**C4 :** Elementary transcendental functions**IMSL subprogram library**

MMLINC Computes an elementary integral from which inverse circular functions, logarithms or inverse hyperbolic functions may be computed by rational operations.

C4a : Trigonometric, inverse trigonometric**CMLIB subprogram library (FNLIB sublibrary)**

CACOS	Arc cosine of complex argument, $\cos^{-1} z$.
CASIN	Arc sin of complex argument, $\sin^{-1} z$.
CATAN	Arc tangent of complex argument, $\tan^{-1} z$.
CATAN2	Quadrant correct arctangent of complex arguments, $\tan^{-1}(z1/z2)$.
CCOT	Cotangent of complex argument, $\cot z$.
COS	Cosine of real argument, $\cos x$.
COSDG	Computes the cosine of an argument given in degrees. Double precision version is DCOSDG.
COT	Cotangent of real argument, $\cot x$. Double precision version is DCOT.
CTAN	Tangent of complex argument, $\tan z$.
DACOS	Arc cosine of double precision argument, $\cos^{-1} d$.
DASIN	Arc sine of double precision argument, $\sin^{-1} d$.
DTAN	Double precision tangent, $\tan d$.
SIN	Sine of real argument, $\sin x$.
SINDG	Computes the Sine of an argument in degrees. Double precision version is DSINDG.

NAG subprogram library

S07AAE	$\tan(x)$. Double precision version is S07AAF.
S09AAE	$\arcsin(x)$. Double precision version is S09AAF.
S09ABE	$\arccos(x)$. Double precision version is S09ABF.

PORT subprogram library

ARCOS	Computes $\arccos(x)$, answer in radians. Double precision version is DARCOS.
ARSIN	Computes $\arcsin(x)$, answer in radians. Double precision version is DARSIN.
TAN	Computes the elementary tangent function. If your Fortran library includes this function, use that instead. Double precision version is DTAN.

C4b : Exponential, logarithmic**CMLIB subprogram library (FNLIB sublibrary)**

ALNREL	$\ln(1 + x)$. Double precision version is DLNREL.
ALOG	$\ln(x)$.
ALOG10	$\log_{10} x$.
CEXPRL	Relative error exponential of complex argument from first order, $(e^z - 1)/z$.
CLNREL	Relative error logarithm of complex argument, $\ln(1+z)$.
CLOG10	Common logarithm of complex argument, $\log_{10} z$.
EXP	Exponential function, $= e^z$.
EXPREL	Relative error exponential from first order, $= (e^z - 1)/z$. Double precision version is DEXPRL.

PORT subprogram library

CDEXP	Computes e^z for complex double precision z . Complex double precision numbers represented as a double precision array of two elements.
CDLOG	Computes $\ln(z)$ for complex double precision z . Complex double precision numbers are represented as

a double precision array of two elements.

C4c : Hyperbolic, inverse hyperbolic

CMLIB subprogram library (FNLIB sublibrary)

ACOSH Hyperbolic cosine $\cosh x$. Double precision version is **DACOSH**.
ASINH Hyperbolic sine $\sinh x$. Double precision version is **DASINH**.
ATANH Arc hyperbolic tangent $\tanh^{-1}x$. Double precision version is **DATANH**.
CACOSH Arc hyperbolic cosine of complex argument, $\cosh^{-1}z$.
CASINH Arc hyperbolic sin of complex argument, $\sinh^{-1}z$.
CATANH Arc hyperbolic tangent of complex argument, $\tanh^{-1}z$.
CCOSH Hyperbolic cosine of complex argument, $\cosh z$.
CSINH Hyperbolic sine of complex argument, $\sinh z$.
CTANH Hyperbolic tangent of complex argument, $\tanh z$.

NAG subprogram library

S10AAE $\tanh(x)$. Double precision version is **S10AAF**.
S10ABE $\sinh(x)$. Double precision version is **S10ABF**.
S10ACE $\cosh(x)$. Double precision version is **S10ACF**.
S11AAE $\operatorname{Arctanh}(x)$. Double precision version is **S11AAF**.
S11ABE $\operatorname{Arcsinh}(x)$. Double precision version is **S11ABF**.
S11ACE $\operatorname{Arccosh}(x)$. Double precision version is **S11ACF**.

PORT subprogram library

ACOSH Computes hyperbolic arccosine, $\operatorname{arccosh}(x)$. Double precision version is **DACOSH**.
ASINH Computes hyperbolic arcsine, $\operatorname{arsinh}(x)$. Double precision version is **DASINH**.
ATANH Computes hyperbolic arctangent, $\operatorname{arctanh}(x)$. Double precision version is **DATANH**.
COSH Computes hyperbolic cosine, $\cosh(x)$. Double precision version is **DCOSH**.
SINH Computes hyperbolic sin, $\sinh(x)$. Double precision version is **DSINH**.
TANH Computes hyperbolic tangent, $\tanh(x)$. Double precision version is **DTANH**.

C4d : Integrals of elementary transcendental functions

C5 : Exponential and logarithmic integrals

CMLIB subprogram library (AMOSLIB sublibrary)

EXINT Computes sequences of exponential integrals $E_{N+K}(x)$, $K=0,\dots,M-1$ or e^x times same to specified tolerance. Double precision version is **DEXINT**.

CMLIB subprogram library (FNLIB sublibrary)

ALI $\int_0^x 1/\ln t \, dt$. Double precision version is **DLI**.
E1 Exponential integral, $\int_x^\infty e^{-t}/t \, dt$. Double precision version is **DE1**.
EI Exponential integral, $\int_{-x}^\infty e^{-t}/t \, dt$. Double precision version is **DEI**.

SPENC Spence Dilogarithm, $-\int_0^x \ln|1-y|/y \, dy$. Double precision version is DSPENC.

IMSL subprogram library

MMDEI Exponential integrals.

MMDEN Exponential integrals of integer order for real argument x scaled by e^x .

MATHWARE subprogram library (STEGUN sublibrary)

EXPINT Computes exponential integral $E_n(x)$. Change computer by changing one line.

SICIEI Computes sine, cosine, exponential integral as well as hyperbolic sin, cosine, exponential integral.

NAG subprogram library

S13AAE Exponential integral, $E_1(x)$. Double precision version is S13AAF.

C6 : Cosine and sine integrals

MATHWARE subprogram library (STEGUN sublibrary)

SICIEI Computes sine, cosine, exponential integral as well as hyperbolic sin, cosine, exponential integral.

NAG subprogram library

S13ACE Cosine integral, $ci(x)$. Double precision version is S13ACF.

S13ADE Sine integral, $si(x)$. Double precision version is S13ADF.

C7 : Gamma

C7a : Gamma, log gamma, reciprocal gamma

CMLIB subprogram library (AMOSLIB sublibrary)

GAMLN Computes $\ln \Gamma(x)$ for non-negative x .

CMLIB subprogram library (FNLIB sublibrary)

ALGAMS $\ln|\Gamma(x)|$, with sign of $\Gamma(x)$. Double precision version is DLGAMS.

ALNGAM $\ln |\Gamma(x)|$. Double precision version is DLNGAM.

CGAMMA Gamma function of complex argument, $\Gamma(z)$.

CGAMR Reciprocal gamma function of complex argument, $1/\Gamma(z)$.

CLNGAM Log gamma of complex argument, $\ln \Gamma(z)$.

GAMMA $\Gamma(z)$. Double precision version is DGAMMA.

GAMR Reciprocal gamma function, $= 1/\Gamma(z)$. Double precision version is DGAMR.

POCH Pochhammer's symbol $(a)_x = \Gamma(a+x)/\Gamma(a)$. Double precision version is DPOCH.

POCH1 Pochhammer's symbol from first order, $= ((a)_x - 1)/x$. Double precision version is DPOCH1.

IMSL subprogram library

ALGAMA Evaluate $\ln|\Gamma(x)|$.

GAMMA Evaluate $\Gamma(x)$.

NAG subprogram library

- S14AAE** Gamma function. Double precision version is S14AAF.
S14ABE Log gamma function. Double precision version is S14ABF.

C7b : Beta, log beta

CMLIB subprogram library (FNLIB sublibrary)

- ALBETA** Log Beta. Double precision version is DLBETA.
BETA $B(a, b) = \Gamma(a)\Gamma(b)/\Gamma(a + b)$. Double precision version is DBETA.
CBETA Beta function of complex arguments, $B(z1, z2) = \Gamma(z1)\Gamma(z2)/\Gamma(z1 + z2)$.
CLBETA Log Beta of complex arguments, $\ln B(z1, z2)$.

C7c : Psi function

CMLIB subprogram library (FNLIB sublibrary)

- CPSI** Psi (digamma) of complex argument, $\psi(z) = \Gamma'(z)/\Gamma(z)$.
PSI Psi (digamma), $\psi(x) = \Gamma'(x)/\Gamma(x)$. Double precision version is DPSI.

IMSL subprogram library

- MMPSI** Logarithmic derivative of the gamma function, $\psi(x) = \Gamma'(x)/\Gamma(x)$.

C7d : Polygamma function

C7e : Incomplete gamma

CMLIB subprogram library (FNLIB sublibrary)

- GAMI** Incomplete gamma function, $\gamma(a, x) = \int_0^x t^{a-1}e^{-t} dt$. Double precision version is DGAMI.
GAMIC Complementary incomplete gamma, $= \int_0^\infty t^{a-1}e^{-t} dt$. Double precision version is DGAMIC.
GAMIT Tricomi's incomplete gamma, $= x^{-a}\gamma(a, x)$. Double precision version is DGAMIT.

C7f : Incomplete beta

CMLIB subprogram library (FNLIB sublibrary)

- BETAI** Incomplete Beta. $I_x(a, b) = B_x(a, b)/B(a, b)$. Double precision version is DBETAI.

C7g : Riemann zeta

C8 : Error functions

C8a : Error functions, their inverses, integrals, including the normal distribution function**CMLIB subprogram library (FNLIB sublibrary)****ERF** Error function, $=2/\sqrt{\pi} \int_0^x \exp(-t^2) dt$. Double precision version is DERF.**ERFC** Complementary error function, $=2/\sqrt{\pi} \int_x^\infty \exp(-t^2) dt$. Double precision version is DERFC.**IMSL subprogram library****ERF** Evaluate the error function. Note: the Fortran mathematical subroutine libraries may also contain ERF.**ERFC** Evaluate the complementary error function.**MDNOR** Normal or Gaussian probability distribution function.**MERFCI** Inverse complementary error function.**MERFI** Inverse error function.**MERRCZ** $\exp(-z^2) \operatorname{erfc}(-iz)$ for complex z .**MATHWARE subprogram library (STEGUN sublibrary)****ERRINT** Computes error function and complementary error function to maximum machine accuracy. To change computers change one line.**NAG subprogram library****S15ABE** Cumulative normal distribution function, $P(x)$. Double precision version is S15ABF.**S15ACE** Complement of cumulative normal distribution function, $Q(x)$. Double precision version is S15ACF.**S15ADE** Complement of error function, $\operatorname{erfc}(x)$. Double precision version is S15ADF.**S15AEE** Error function, $\operatorname{erf}(x)$. Double precision version is S15AEF.**C8b :** Fresnel integrals**NAG subprogram library****S20ACE** Fresnel integrals, $S(x)$. Double precision version is S20ACF.**S20ADE** Fresnel integrals, $C(x)$. Double precision version is S20ADF.**C8c :** Dawson's integral**CMLIB subprogram library (FNLIB sublibrary)****DAWS** Dawson's function, $F(x) = \exp(-x^2) \int_0^x \exp(t^2) dt$. Double precision version is DDAWS.**IMSL subprogram library****MMDAS** Dawson integral.**NAG subprogram library****S15AFE** Dawson's integral. Double precision version is S15AFF.

C9 : Legendre functions**CMLIB subprogram library (FCNPAK sublibrary)**

XSLEGF Calculates sequences of values of Legendre functions of the first kind of positive or negative order, as well as Legendre functions of the second kind, and normalized Legendre functions. Double precision version is XDLEGF.

XSNRMP Calculates a sequence of values of the normalized Legendre polynomials for fixed degree and argument and variable order. Double precision version is XDNRMP.

C10 : Bessel functions**C10a :** J , Y , $H^{(1)}$, $H^{(2)}$ **C10a1 :** Real argument, integer order**CMLIB subprogram library (FNLIB sublibrary)**

BESJ0 $J_0(x)$. Double precision version is DBESJ0.

BESJ1 $J_1(x)$. Double precision version is DBESJ1.

BESY0 $Y_0(x)$. Double precision version is DBESY0.

BESY1 $Y_1(x)$. Double precision version is DBESY1.

IMSL subprogram library

MMBSJ0 $J_0(x)$.

MMBSJ1 $J_1(x)$.

MMBSJN $J_n(x)$ $n = 0, 1, \dots, N$

NAG subprogram library

S17ACE $Y_0(x)$. Double precision version is S17ACF.

S17ADE $Y_1(x)$. Double precision version is S17ADF.

S17AEE $J_0(x)$. Double precision version is S17AEF.

S17AFE $J_1(x)$. Double precision version is S17AFF.

PORT subprogram library

BESRJ $J_n(x)$ $n = 0, 1, \dots, N$. Double precision version is DBESRJ.

C10a2 : Complex argument, integer order**CMLIB subprogram library (AMOSLIB sublibrary)**

CJYHBS Computes $J_0, J_1, Y_0, Y_1, H_0, H_1$ (Bessel and Struve functions) of complex argument.

IMSL subprogram library

MMBZJN $J_n(z)$, $n = 0, 1, \dots, N$.

PORT subprogram library

BES CJ $J_n(z)$, $n = 0, 1, \dots, N$. Double precision version is DBES CJ.

C10a3 : Real argument, real order

CMLIB subprogram library (AMOSLIB sublibrary)

BESJ $J_{\alpha+n}(x)$ $n = 1, \dots, N$, $\alpha, x \geq 0$. Uses internal double precision arithmetic.

IMSL subprogram library

MMBSJR $J_{n+\alpha}(x)$, $n = 0, 1, \dots, N$, $\alpha, x \geq 0$.

MMBSYN $Y_{n+\alpha}(x)$, $n = 0, 1, \dots, N$, $\alpha, x \geq 0$.

C10a4 : Complex argument, real order

C10a5 : Complex argument, complex order

C10b : I, K

C10b1 : Real argument, integer order

CMLIB subprogram library (FNLIB sublibrary)

BESIO $I_0(x)$. Double precision version is DBESIO.

BESIOE $\exp(-|x|)I_0(x)$. Double precision version is DBSIOE.

BESI1 $I_1(x)$. Double precision version is DBESI1.

BESI1E $\exp(-|x|)I_1(x)$. Double precision version is DBSI1E.

BESK0 $K_0(x)$. Double precision version is DBESK0.

BESK0E $\exp(x)K_0(x)$. Double precision version is DBSK0E.

BESK1 $K_1(x)$. Double precision version is DBESK1.

BESK1E $\exp(x)K_1(x)$. Double precision version is DBSK1E.

IMSL subprogram library

MMBSIO $I_0(x)$.

MMBSI1 $I_1(x)$.

MMBSIN $I_n(x)$, $n = 0, 1, \dots, N$.

MMBSK0 $K_0(x)$.

MMBSK1 $K_1(x)$.

NAG subprogram library

S18ACE $K_0(x)$. Double precision version is S18ACF.

S18ADE $K_1(x)$. Double precision version is S18ADF.

S18AEE $I_0(x)$. Double precision version is S18AEF.

S18AFE $I_1(x)$. Double precision version is S18AFF.

S18CCE $\exp(x)K_0(x)$. Double precision version is S18CCF.

S18CDE $\exp(x)K_1(x)$. Double precision version is S18CDF.

S18CEE $\exp(-|x|)I_0(x)$. Double precision version is S18CEF.

S18CFE $\exp(-|x|)I_1(x)$. Double precision version is S18CFF.

PORT subprogram library

BESRI $I_n(x)$, $n = 0, 1, \dots, N$. Double precision version is DBESRI.

C10b2 : Complex argument, integer order

IMSL subprogram library

MMBZIN $I_n(z)$, $n = 0, 1, \dots, N$.

PORT subprogram library

BESCI $I_n(z)$, $n = 0, 1, \dots, N$. Double precision version is DBESCI.

C10b3 : Real argument, real order

CMLIB subprogram library (FNLIB sublibrary)

BESKES $\exp(x)K_{n+\alpha}(x)$, $n = 0, 1, \dots, N$ if $N \geq 0, -1 < \alpha < 1$, or $\exp(x)K_{\alpha-n}(x)$, $n = 0, 1, \dots, |N|$ if $N < 0, -1 < \alpha < 1$. Double precision version is DBSKES.

BESKS $K_{n+\alpha}(x)$, $n = 0, 1, \dots, N$ if $N \geq 0, -1 < \alpha < 1$, or $K_{\alpha-n}(x)$, $n = 0, 1, \dots, |N|$ if $N < 0, -1 < \alpha < 1$. Double precision version is DBESKS.

IMSL subprogram library

MMBSIR $I_{n+\alpha}(x)$, $n = 0, 1, \dots, N$, $\alpha, x \geq 0$. Exponential scaling optional.

MMBSKR $K_{n+\alpha}(x)$, $n = 0, 1, \dots, N$, $\alpha, x \geq 0$. Exponential scaling optional.

C10b4 : Complex argument, real order

C10b5 : Complex argument, complex order

C10c : Kelvin functions

IMSL subprogram library

MMKEL0 $\text{ber}_0(x)$, $\text{bei}_0(x)$, $\text{ker}_0(x)$, $\text{kei}_0(x)$.

MMKEL1 $\text{ber}_1(x)$, $\text{ber}_1(x)$, $\text{ker}_1(x)$, $\text{kei}_1(x)$.

MMKELD $\text{ber}'_0(x)$, $\text{bei}'_0(x)$, $\text{ker}'_0(x)$, $\text{kei}'_0(x)$.

C10d : Airy and Scorer functions

CMLIB subprogram library (FNLIB sublibrary)

AI $Ai(x)$. Double precision version is DAI.

AIE $Ai(x)$, exponentially scaled. Double precision version is DAIE.

BI $Bi(x)$. Double precision version is DBI.

BIE $Bi(x)$, exponentially scaled. Double precision version is DBIE.

NAG subprogram library

S17AGE $Ai(z)$. Double precision version is S17AGF.
S17AHE $Bi(z)$. Double precision version is S17AHF.
S17AJE $Ai'(z)$. Double precision version is S17AJF.
S17AKE $Bi'(z)$. Double precision version is S17AKF.

C10e : Struve, Anger, and Weber functions

CMLIB subprogram library (AMOSLIB sublibrary)

CJYHBS Computes $J_0, J_1, Y_0, Y_1, H_0, H_1$ (Bessel and Struve functions) of complex argument.

C10f : Integrals of Bessel functions

C11 : Confluent hypergeometric functions

CMLIB subprogram library (FNLIB sublibrary)

CHU $U(a, b, z)$. Double precision version is DCHU.

C12 : Coulomb wave functions

C13 : Jacobian elliptic functions, theta functions

C14 : Elliptic integrals

IMSL subprogram library

MMDELE Complete elliptic integral of the second kind.
MMDELK Complete elliptic integral of the first kind.
MMLIND Complete or incomplete elliptic integral of the second kind.
MMLINF Complete or incomplete elliptic integral of the first kind.
MMLINJ Complete or incomplete elliptic integral of the third kind.

NAG subprogram library

S21BAE Degenerate elliptic integrals of 1st kind, primarily for use in computing nondegenerate elliptic integrals and certain elementary circular and hyperbolic functions. Double precision version is S21BAF.
S21BBE Elliptic integrals of 1st kind (complete or incomplete). Double precision version is S21BBF.
S21BCE Elliptic integrals of 2nd kind (complete or incomplete). Double precision version is S21BCF.
S21BDE Elliptic integrals of 3rd kind (complete or incomplete). Double precision version is S21BDF.

CMLIB subprogram library (FCNPAK sublibrary)

RC Degenerate elliptic integrals of 1st kind, primarily for use in computing nondegenerate elliptic integrals and certain elementary circular and hyperbolic functions.

RF	Elliptic integrals of 1st kind (complete or incomplete).
RD	Elliptic integrals of 2nd kind (complete or incomplete).
RJ	Elliptic integrals of 3rd kind (complete or incomplete).

C15 :	Weierstrass elliptic functions
--------------	--------------------------------

IMSL subprogram library

MMWPL	Weierstrass p-function in the lemniscatic case for complex argument with unit period parallelogram.
MMWPL1	First derivative of the Weierstrass p-function in the lemniscatic case for complex argument with unit period parallelogram.
MMWPQ	Weierstrass p-function in the equianharmonic case for complex argument with unit period parallelogram.
MMWPQ1	First derivative of the Weierstrass p-function in the equianharmonic case for complex argument with unit period parallelogram.

C16 :	Parabolic cylinder functions
--------------	------------------------------

C17 :	Mathieu functions
--------------	-------------------

C18 :	Spheroidal wave functions
--------------	---------------------------

C19 :	Other special functions
--------------	-------------------------

D: Linear Algebra

The most widely used constructs in mathematical computation are, undoubtedly, matrices and vectors. Consequently, it is not surprising to find a rich selection of software designed to perform operations such as computing solutions of linear systems, eigenvalues, determinants, and inverses. While it may seem that there are many routines that perform essentially the same task, one will find that many of these routines are designed to take advantage of special features of a given problem. Thus, to select the most appropriate routine one should determine as many attributes of one's matrix as possible. Some important attributes are given below.

1. The entries of the matrix are either **real** or **complex** numbers.
2. Special symmetry properties are important. If the matrix is real, it is called **symmetric** when the elements satisfy $a_{ji} = a_{ij}$. If the matrix is complex, it is called **Hermitian** when the elements a_{ij} of the matrix A satisfy $a_{ji} = \text{complex conjugate of } a_{ij}$.
3. The location and density of the non-zero entries of the matrix are important attributes. With no other knowledge of the matrix one would say the matrix is **dense**, i.e. that most of the entries are non-zero. A matrix is called **sparse** if there are relatively few non-zero entries. The cross-over point between dense and sparse is difficult to determine in practice.
4. Of importance among sparse matrices are **band** matrices wherein all the non-zero entries are concentrated on a band near the main diagonal of the matrix. Such matrices commonly arise when differential equations are discretized. **Tridiagonal** matrices are band matrices those in which the band consists only of the main diagonal and the closest lower and upper diagonals.
5. Hermitian and symmetric matrices may also have another important attribute—**positive definiteness**. A matrix A is positive definite if $x^T A x > 0$ for all $x > 0$. This condition may be difficult to determine in practice without some independent knowledge of the matrix. See one of the references for more information on this attribute.

Most of the software classified in chapter D are Fortran subprograms. The notable exception is MATLAB, an interactive system for matrix analysis. In MATLAB one can solve linear systems of equations, determine eigenvalues, compute inverses, and solve least squares problems using simple interactive commands which operate on vectors and matrices.

January 1984

References

1. Duff, I. S. (1977). A Survey of Sparse Matrix Research, *Proceedings of the IEEE*, Vol. 65, No. 4, pp. 500-535.
2. Noble, B. and J. W. Daniel (1977). *Applied Linear Algebra*, 2nd Edition, Prentice-Hall, Englewood Cliffs, New Jersey.
3. Reid, J. K. (1977). Sparse Matrices, in *The State of the Art in Numerical Analysis*, D. Jacobs (Editor), Academic Press, New York, pp. 85-146.
4. Stewart, G. W. (1973). *Introduction to Matrix Computations*, Academic Press, New York.

D : Linear Algebra

MATLAB interactive system

- **MATLAB** An interactive system for defining and manipulating matrices. It includes solving linear systems, linear least squares, eigenvalue and eigenvector calculation, QR decomposition, singular value decomposition and inverses, as well as other interesting and useful features.

MINITAB interactive system

- MINITAB** Minitab's vector and matrix commands include COPY (vectors, vectors to matrices, and conversely), DIAGONAL (create a diagonal matrix or extract the diagonal of a matrix), TRANSPOSE, INVERSE, and EIGEN (calculate eigenvalues and eigenvectors for a symmetric matrix).

D1: Elementary vector and matrix operations

The subprograms in this class provide automatic computation of common elementary vector and matrix operations, such as the dot (inner) product of two vectors or the product of two matrices. These are provided to reduce the time and effort required to code problems. In addition, they can also help in making a program more portable while at the same time achieving high efficiency on a wide variety of machines. For example, the Basic Linear Algebra Subprograms (BLAS) available in CMLIB are widely distributed and are often implemented in machine language to increase their efficiency. Since these operations are often found in the innermost loops of large computations the resulting savings can often be significant.

January 1984

References

1. Lawson, C. L., R. J. Hanson, D. R. Kincaid, et al. (1979). Basic Linear Algebra Subprograms for Fortran Usage, *ACM Transactions on Mathematical Software*, Vol 5., No. 3, pp. 308-323.

D1 :	Elementary vector and matrix operations
-------------	---

D1a :	Elementary vector operations
--------------	------------------------------

D1a1 :	Set to constant
---------------	-----------------

DATAPAC subprogram library

DEFINE Defines a vector of constants by setting all of the elements in the single precision vector X equal to XNEW.

NAG subprogram library

F01CQE Sets the elements of a vector to zero. Double precision version is F01CQF.

PORT subprogram library

SETC Set a specified number of values in a complex array equal to a constant.

SETI Set a specified number of values in an integer array equal to a constant.

SETL Set a specified number of values in a logical array equal to a constant.

SETR Set a specified number of values in a real array equal to a constant. Double precision version is SETD.

D1a2 :	Minimum and maximum components
---------------	--------------------------------

CMLIB subprogram library (BLAS sublibrary)

ICAMAX Find smallest index of maximum magnitude component of a complex vector.

ISAMAX Find smallest index of maximum magnitude component of a single precision vector. Double precision version is IDAMAX.

CMLIB subprogram library (XBLAS sublibrary)

ISAMIN Find the smallest index of the minimum magnitude component of a real vector.

ISMAX Find the smallest index of the maximum component of a real vector.

ISMIN Find the smallest index of the minimum component of a real vector.

DATAPAC subprogram library

MAX Computes the sample maximum of the data in the input vector X.

MIN Computes the sample minimum of the data in the input vector X.

IMSL subprogram library

USMNMX Determination of the minimum and maximum values of a vector.

PORT subprogram library

EXTRMI Finds extremal points of an integer function defined on a mesh.

EXTRMR Finds extremal points of a real function defined on a mesh. Double precision version is EXTRMD.

D1a3 : Norm

D1a3a : L-1 (sum of magnitudes)

CMLIB subprogram library (BLAS sublibrary)

SASUM Compute single precision sum of absolute values of components of vector. Double precision version is DASUM.

SCASUM Compute complex sum of absolute values of components of vector.

IMSL subprogram library

VABSMF Sum of the absolute values of the elements of a vector or a subset of a vector.

VABSMS Sum of the absolute values of the elements of a row (or column) of a matrix stored in symmetric storage mode.

D1a3b : L-2 (Euclidean norm)

CMLIB subprogram library (BLAS sublibrary)

SCNRM2 Compute the Euclidean length or L2 norm of a complex vector.

SNRM2 Compute the Euclidean length or L2 norm of a single precision vector. Double precision version is DNRM2.

NAG subprogram library

F05ABE Approximate 2-norm of a vector. Double precision version is F05ABF.

PORT subprogram library

SNRM2 Finds the length (Euclidean norm) of a vector, without underflow or overflow. Double precision version is DNRM2.

D1a3c : L-infinity (maximum magnitude)

IMSL subprogram library

VABMXF Maximum absolute value of the elements of a vector or a subset of the elements of a vector.

VABMXS Maximum absolute value of the elements of a row or column of a matrix stored in symmetric storage mode.

D1a4 : Dot product (inner product)**CMLIB subprogram library (BLAS sublibrary)**

- CDOTC** Compute complex dot product using conjugated vector components.
- CDOTU** Compute complex dot product using unconjugated vector components.
- DQDOTA** Compute dot product of 2 double precision vectors plus a double precision constant plus extended precision constant.
- DQDOTI** Compute in extended precision dot product of 2 double precision vectors plus d.p. constant. Result is double precision.
- DSDOT** Compute single precision dot product using double precision accumulation.
- SDOT** Compute single precision dot product. Double precision version is DDOT.
- SDSDOT** Compute single precision dot product and add a constant using double precision accumulation.

CMLIB subprogram library (XBLAS sublibrary)

- CDCDOT** Computes complex precision dot product and adds a scalar. Uses double precision accumulation.
- DCDOT** Computes a complex precision dot product using double precision accumulation.

IMSL subprogram library

- VIPRFF** Vector inner product of two vectors or subsets of two vectors.
- VIPRSS** Vector inner product of two vectors each of which is part of some matrix stored in symmetric mode.

NAG subprogram library

- F01DAE** Returns the sum of an initial value and a scalar product, using basic precision arithmetic. Double precision version is F01DAF.
- F01DBE** Returns the sum of an initial value and a scalar product, using additional precision arithmetic. Double precision version is F01DBF.
- F01DCE** Computes the value of a complex scalar product and subtracts it from a complex initial value, using basic precision arithmetic. Double precision version is F01DCF.
- F01DDE** Computes the value of a complex scalar product and subtracts it from a complex initial value, using additional precision arithmetic. Double precision version is F01DDF.
- F01DEE** Returns the value of the scalar product of two arrays of length N, using basic precision arithmetic. Double precision version is F01DEF.
- X03AAE** Real innerproduct added to initial value, basic/additional precision. Double precision version is X03AAF.
- X03ABE** Complex innerproduct added to initial value, basic/additional precision. Double precision version is X03ABF.

D1a5 : Copy or exchange (swap)**CMLIB subprogram library (BLAS sublibrary)**

- CCOPY** Copy a vector X to a vector Y, both complex.
- CSWAP** Interchange vectors X and Y, both complex.
- SCOPY** Copy a vector X to a vector Y, both single precision. Double precision version is DCOPY.
- SSWAP** Interchange vectors X and Y, both single precision. Double precision version is DSWAP.

CMLIB subprogram library (XBLAS sublibrary)

- SCOPYM** Copies negative of array SX into array SY, with corresponding increments INCX and INCY.

DATAPAC subprogram library

COPY Copies the contents of the vector X into vector Y.
MOVE Copies M elements of the vector X (starting with position IX1) into the vector Y (starting with position IY1).

NAG subprogram library

F01CNE Copies a vector of length M into a row of a matrix. Double precision version is F01CNF.
F01CPE Copies the contents of a vector into a second vector. Double precision version is F01CPF.

PORT subprogram library

MOVEBC Move a complex vector using backward DO loop.
MOVEBI Move an integer vector using backward DO loop.
MOVEBL Move a logical vector using backward DO loop.
MOVEBR Move a real vector using backward DO loop. Double precision version is MOVEBD.
MOVEFC Move a complex vector using forward DO loops.
MOVEFD Move a double precision vector using forward DO loop.
MOVEFI Move an integer vector using forward DO loop.
MOVEFL Move a logical vector using forward DO loop.
MOVEFR Move a real vector using forward DO loop. Double precision version is MOVFRD.

D1a6 : Multiplication by scalar

CMLIB subprogram library (BLAS sublibrary)

CSCAL Compute complex constant times complex vector.
CSSCAL Compute real constant times complex vector.
SSCAL Compute a constant times a vector, both single precision. Double precision version is DSCAL.

MATHWARE subprogram library (NASHLIB sublibrary)

A11VS Standardizes a complex vector to have maximum component of magnitude equal to one.

D1a7 : Triad ($a*x+y$ for vectors x,y and scalar a)

CMLIB subprogram library (BLAS sublibrary)

CAXPY Compute a constant times a vector plus a vector, all complex.
SAXPY Compute a constant times a vector plus a vector, all single precision. Double precision version is DAXPY.

D1a8 : Elementary rotation (Givens transformation)

CMLIB subprogram library (BLAS sublibrary)

SROT Apply Givens plane rotation to a single precision vector. Double precision version is DROT.
SROTM Apply modified Givens plane rotation to single precision vector. Double precision version is DROTM.

D1a9 : Elementary reflection (Householder transformation)

IMSL subprogram library

VHS12 Real Householder transformation computation and applications.

D1a10 : Convolutions

IMSL subprogram library

VCONVO Vector convolution.

NAG subprogram library

C06ACE Circular convolution of two real vectors of period 2^m . Double precision version is C06ACF.

D1b : Elementary matrix operations

D1b1 : Set to zero, to identity

NAG subprogram library

F01CAE Sets elements of an mn matrix A to zero. Double precision version is F01CAF.

F01CBE Sets the elements $A(i,j)$ to one if $i=j$ and zero otherwise, where $1 \leq i \leq m$ and $1 \leq j \leq n$. Double precision version is F01CBF.

D1b2 : Norm

IMSL subprogram library

VNRMF1 1-norm of matrices (full storage mode).

VNRMF2 Euclidean-norm of matrices (full storage mode).

VNRMFI Infinity-norm matrices (full storage mode).

VNRMS1 1-norm of matrices (symmetric storage mode).

VNRMS2 Euclidean-norm of matrices (symmetric storage mode).

D1b3 : Transpose

IMSL subprogram library

VTRAN Transpose a rectangular matrix.

NAG subprogram library

F01CRE Re-orders the elements of a vector of length mn , containing an mn matrix, A, so that the new vector contains the transpose matrix. Double precision version is F01CRF.

D1b4 : Multiplication by vector**NAG subprogram library**

F01CSE Forms the product $c=Ab$ where b is a vector and A is a symmetric matrix whose lower triangle is stored by rows in a one-dimensional array. Double precision version is F01CSF.

D1b5 : Addition, subtraction**IMSL subprogram library**

VUABQ Matrix addition (band + band symmetric matrices).
VUAFB Matrix addition (full + band matrices).
VUAFQ Matrix addition (full + band symmetric matrices).
VUAFS Matrix addition (full + symmetric matrices).
VUASB Matrix addition (symmetric + band matrices).
VUASQ Matrix addition (symmetric + band symmetric matrices).

NAG subprogram library

F01CDE Adds elements of the mn matrices B and C and stores the results in elements of the matrix A . Double precision version is F01CDF.
F01CEE Subtracts elements of the matrix C from elements of the matrix B and stores the results in elements of the matrix A . Double precision version is F01CEF.
F01CGE Adds elements of the matrix B to elements in different positions in the matrix A . Double precision version is F01CGF.
F01CHE Subtracts elements of the matrix B from elements in a different position in the matrix A . Double precision version is F01CHF.

D1b6 : Multiplication**IMSL subprogram library**

VMULBB Matrix multiplication (band storage mode).
VMULBF Matrix multiplication (band by full matrices).
VMULBS Matrix multiplication (band by symmetric matrices).
VMULFB Matrix multiplication (full by band matrices).
VMULFF Matrix multiplication (full storage mode).
VMULFM Matrix multiplication of the transpose of matrix a by matrix b (full storage mode).
VMULFP Matrix multiplication of matrix a by the transpose of matrix b (full storage mode).
VMULFQ Matrix multiplication (full by band symmetric matrices).
VMULFS Matrix multiplication (full by symmetric matrices).
VMULQB Matrix multiplication (band symmetric by band matrices).
VMULQF Matrix multiplication (band symmetric by full matrices).
VMULQQ Matrix multiplication (band symmetric storage mode).
VMULQS Matrix multiplication (band symmetric by symmetric matrices).
VMULSB Matrix multiplication (symmetric by band matrices).
VMULSF Matrix multiplication (symmetric by full matrices).

VMULSQ Matrix multiplication (symmetric by band symmetric matrices).
VMULSS Matrix multiplication (symmetric storage mode).
VTPROF Transpose product of matrix (full storage mode).
VTPROS Transpose product of a matrix (symmetric storage mode).

NAG subprogram library

F01CKE Returns with the result of multiplication of two matrices B and C in the matrix A, with the option to overwrite B or C. Double precision version is F01CKF.
F01CLE Post-multiplies the matrix B with the transpose of the matrix C and places the result in the matrix A. Double precision version is F01CLF.

D1b7 : Matrix polynomial

IMSL subprogram library

VPOLYF Matrix polynomial (full storage mode).

D1b8 : Copy

NAG subprogram library

F01CFE Copies elements of the matrix B into different positions in the matrix A. Double precision version is F01CFF.
F01CME Copies elements of one matrix into a second matrix. Double precision version is F01CMF.
F01CNE Copies a vector of length M into a row of a matrix. Double precision version is F01CNF.

D1b9 : Storage mode conversion

IMSL subprogram library

VCVTBF Storage mode conversion of matrices (band to full storage mode).
VCVTCH Storage mode conversion of matrices (full complex to Hermitian).
VCVTFB Storage mode conversion of matrices (full to band storage mode).
VCVTFQ Storage mode conversion (full to band symmetric storage mode).
VCVTFS Storage mode conversion of matrices (full to symmetric).
VCVTHC Storage mode conversion of matrices (Hermitian to full complex).
VCVTQF Storage mode conversion (band symmetric to full storage mode).
VCVTQS Storage mode conversion (band symmetric to symmetric storage mode).
VCVTSF Storage mode conversion of matrices (symmetric to full).
VCVTSQ Storage mode conversion (symmetric to band symmetric storage mode).

D1b10 : Elementary rotation (Givens transformation)

CMLIB subprogram library (BLAS sublibrary)

CROTG Construct Givens plane rotation of complex matrix.

SROTG Construct Givens plane rotation of single precision matrix. Double precision version is DROTG.
SROTMG Construct modified Givens plane rotation of single precision matrix. Double precision version is DROTMG.

CMLIB subprogram library (XBLAS sublibrary)

CSROT Applies Givens plane rotation to complex matrix.

MATHWARE subprogram library (NASHLIB sublibrary)

ROTN Plane rotation of two rows of a matrix.

D1b11 : Elementary reflection (Householder transformation)

IMSL subprogram library

VHS12 Real Householder transformation computation and applications.
VHSH2C Complex Householder transformation to zero a single element of a matrix.
VHSH2R Real Householder transformation to zero a single element of a matrix.
VHSH3R Real Householder transformation to zero two elements of a matrix.

**D2: Solutions of systems of linear equations
 (including inversion, LU, and related decompositions)**

Software in this class is concerned with the solution of nonsingular systems of linear equations

$$Ax = b$$

for an unknown vector x . (Software for solving problems where the solution is overdetermined or underdetermined, i.e., A is not square, can be found in class D9.) Matrix inversion is almost never the correct way to solve this problem. Routines specially designed for solving such systems are much to be preferred, both for speed and accuracy. On reflection one often will discover that the expression $A^{-1}b$ is simply a convenient way of writing "the solution of the system $Ax = b$ " and that one should do just that rather than form an inverse explicitly. One exception is the calculation of a matrix inverse in order to obtain covariances for some statistical applications.

Since operations involving matrices frequently form a major part of the computation time for a given problem, great effort has been expended to isolate the dominant part of a particular computation into a single routine in order to avoid unnecessary repetitions when possible. This has the somewhat annoying consequence that one may be required to call two routines to solve a single problem. However, in some cases this isolation can significantly reduce the computation time. As an example, suppose we want to solve the linear system above for many different vectors b . The procedure for solving would be two steps:

- (a) the matrix A is factored as a product of two or three matrices of a special form independent of the vector b , and
- (b) this factorization is used to solve for x given b .

If A is of order n , then generally step (a) requires computational time proportional to n^3 , whereas step (b) requires only a time proportional to n^2 . Hence, if one has many different right side vectors b , it is clearly more efficient to do step (a) only once. This situation occurs, for example, when solving time-dependent partial differential equations by implicit methods.

The Fortran subprogram libraries CMLIB, IMSL, and NAG each have large selections of programs for solving this problem. The principal public-domain package is LINPACK, which was developed at Argonne National Laboratory. Both LINPACK and a set of driver routines, called LINDRIVES, are available in CMLIB. These routines are classified according to the type of matrix A to which they may be applied, so it is important to determine as much about the special structure of a problem as possible in order to choose the best routine.

January 1984

References

1. Dongarra, J. J., J. R. Bunch, C. B. Moler, et al. (1979). *LINPACK Users' Guide*, SIAM, Philadelphia.
2. Forsythe, G. and C. Moler (1967). *Computer Solution of Linear Algebraic Systems*, Prentice-Hall, Englewood Cliffs, New Jersey.

3. George, A. and J. W. H. Liu (1981). *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, New Jersey.

4. Hageman, L. A. and D. M. Young (1981). *Applied Iterative Methods*, Academic Press, New York.

5. Young, D. M. (1971). *Iterative Solution of Large Linear Systems*, Academic Press, New York.

D2 : Solution of systems of linear equations (including inversion, LU and related decompositions)

D2a : Real nonsymmetric matrices

D2a1 : General

CMLIB subprogram library (LINDRIVES sublibrary)

• **SGEFS** Factors and solves a general NXN single precision system of linear equations. Double precision version is DGEFS.

SGEIR Factors and solves a general single precision system of linear equations and estimates solution accuracy (requires NxN extra storage).

CMLIB subprogram library (LINPACKS sublibrary)

SGECO Computes LU factorization of real general matrix and estimates its condition. Double precision version is DGECCO.

SGEDI Uses LU factorization of real general matrix to compute its determinant and/or inverse. Double precision version is DGEDL.

SGEFA Computes LU factorization of real general matrix. Double precision version is DGEFA.

SGESL Uses LU factorization of real general matrix to solve systems. Double precision version is DGESL.

SQRSL Applies the output of SQRDC to compute coordinate transformations, projections, and least squares solutions (general real matrix). Double precision version is DQRSL.

IMSL subprogram library

LEQIF Linear equation solution - full matrices (virtual memory version).

LEQOF Linear equation solution - full matrices (out-of-core version).

LEQT1F Linear equation solution - full storage mode - space economizer solution.

LEQT2F Linear equation solution - full storage mode - high accuracy solution.

LINV1F Inversion of a matrix - full storage mode space economizer solution.

LINV2F Inversion of a matrix - full storage mode high accuracy solution.

LINV3F In place inverse, equation solution, and/or determinant evaluation - full storage mode.

LUDATF L-U decomposition by the Crout algorithm with optional accuracy test.

LUELMF Elimination part of solution of $Ax=b$ (full storage mode).

LUREFF Refinement of solution to linear equations full storage mode.

MATHWARE subprogram library (NASHLIB sublibrary)

A5GE Gauss elimination with partial pivoting for solution of system of linear equations, $Ax=f$.

A6BS Back substitution for the solution of a triangular system of linear equations, $Rx=f$.

NAG subprogram library

F01AAE Calculates the approximate inverse of a real matrix by Crout's method with partial pivoting. Double precision version is F01AAF.

- F01BTE** Decomposes a real matrix into product of triangular matrices LU by Gaussian elimination with partial pivoting. Block-column method used for efficiency on paged virtual machines. Double precision version is F01BTF.
- F03AFE** LU-factorisation and determinant, real matrix. Double precision version is F03AFF.
- **F04AAE** Simultaneous linear equations (black box), real matrix, approximate solution, multiple right hand sides. Double precision version is F04AAF.
- **F04AEE** Simultaneous linear equations (black box), real matrix, accurate solution, multiple right hand sides. Double precision version is F04AEF.
- F04AHE** Calculates the accurate solution of set of real linear equations with multiple right hand sides, $AX=B$, where A has been decomposed into triangular matrices using F03AFE. Double precision version is F04AHF.
- F04AJE** Calculates the approximate solution of set of real linear equations with multiple right hand sides, $AX=B$, where A has been decomposed into triangular matrices using F03AFE. Double precision version is F04AJF.
- **F04ARE** Simultaneous linear equations (black box), real matrix, approximate solution, one right hand side. Double precision version is F04ARF.
- **F04ATE** Simultaneous linear equations (black box), real matrix, accurate solution, one right hand side. Double precision version is F04ATF.
- F04AYE** Calculates the approximate solution of set of real linear equations with multiple right hand sides, $AX=B$, where A has been decomposed into triangular matrices using F01BTE. Double precision version is F04AYF.

PORT subprogram library

- LINEQ** Solves a real system of linear equations, $AX=B$, where B is allowed to be a matrix or a vector. Double precision version is DLINEQ.

D2a2 : Banded

CMLIB subprogram library (LINDRIVES sublibrary)

- SNBCO** Factors a real band matrix by Gaussian elimination and estimates condition of the matrix. Double precision version is DNBCO.
- SNBFA** Factors a single precision band matrix by elimination. Double precision version is DNBFA.
- **SNBFS** Factors and solves a general nonsymmetric single precision banded system of linear equations. Double precision version is DNBFS.
- SNBIR** Factors and solves a general nonsymmetric single precision banded system of equations and estimates solution accuracy (needs $N \times (2ML + MU)$ extra storage).
- SNBSL** Solves a general nonsymmetric single precision banded system of linear equations using factors computed previously. Double precision version is DNBSL.

CMLIB subprogram library (LINPACKS sublibrary)

- SGBCO** Computes LU factorization of real band matrix and estimates its condition. Double precision version is DGBCO.
- SGBFA** Computes LU factorization of real band matrix. Double precision version is DGBFA.
- SGBSL** Uses LU factorization of real band matrix to solve systems. Double precision version is DGBSL.

IMSL subprogram library

- LEQT1B** Linear equation solution - band storage mode - space economizer solution.
- LEQT2B** Linear equation solution - band storage mode - high accuracy solution.

NAG subprogram library

- F01LBE** LU-factorisation, real band matrix. Double precision version is F01LBF.
- F04LDE** Simultaneous linear equations (factorising the matrix of coefficients), real band matrix, approximate solution. Double precision version is F04LDF.

D2a2a : Tridiagonal

CMLIB subprogram library (LINPACKS sublibrary)

- SGTSL** Factors a real tridiagonal matrix and simultaneously solves a system. Double precision version is DGTSL.

D2a3 : Triangular

CMLIB subprogram library (LINPACKS sublibrary)

- STRCO** Estimates the condition of real triangular matrix. Double precision version is DTRCO.
- STRDI** Computes determinant and/or inverse of real triangular matrix. Double precision version is DTRDI.
- STRSL** Solves systems with real triangular matrix. Double precision version is DTRSL.

D2a4 : Sparse

CMLIB subprogram library (FSHPK sublibrary)

- POIS3D** Solves block tridiagonal linear systems of algebraic equations arising from the discretization of separable elliptic partial differential equations in 3D.
- POISTG** Solves block tridiagonal linear systems of algebraic equations arising from the discretization of separable elliptic partial differential equations.

CMLIB subprogram library (SLVBLK sublibrary)

- SLVBLK** Solves $Ax=b$ where A is an almost block diagonal matrix. These arise in finite element or piecewise polynomial approximation.

CMLIB subprogram library (YSMP sublibrary)

- CDRV** Solves sparse systems of linear algebraic equations by Gaussian elimination without pivoting (compressed storage mode).
- NDRV** Solves sparse systems of linear algebraic equations by Gaussian elimination without pivoting (uncompressed storage mode).
- TDRV** Solves sparse systems of linear algebraic equations by Gaussian elimination without pivoting.

MATHWARE subprogram library (ITPACK sublibrary)

- JCG** Iterative solution of large sparse systems of linear equations. Jacobi method, conjugate gradient acceleration, adaptive parameter selection.
- JSI** Iterative solution of large sparse systems of linear equations. Jacobi method, Chebyshev acceleration, adaptive parameter selection.
- RSCG** Iterative solution of large sparse systems of linear equations. Reduced system method, conjugate gradient acceleration, adaptive.
- RSSI** Iterative solution of large sparse systems of linear equations. Reduced system method, Chebyshev acceleration, adaptive.

- SOR** Iterative solution of large sparse systems of linear equations. SOR method, adaptive parameter selection.
- SSORCG** Iterative solution of large sparse systems of linear equations. SSOR method, conjugate gradient acceleration, adaptive parameter selection.
- SSORSI** Iterative solution of large sparse systems of linear equations. SSOR method, Chebyshev acceleration, adaptive parameter selection.

NAG subprogram library

- F01BRE** Decomposes real sparse matrix. Either forms LU-decomposition of permutation of entire matrix, or permutes matrix to block lower triangular form and then decomposes diagonal blocks. Double precision version is F01BRF.
- F01BSE** Decomposes a real sparse matrix using the pivotal sequence previously obtained by F01BRE when a matrix of the same sparsity pattern was decomposed. Double precision version is F01BSF.
- F04AXE** Calculates the approximate solution of a set of real sparse linear equations with a single right hand side, $Ax = b$ or $At x = b$, where A has been decomposed by F01BRE or F01BSE. Double precision version is F04AXF.

D2b :	Real symmetric matrices
--------------	-------------------------

D2b1 :	General
---------------	---------

D2b1a :	Indefinite
----------------	------------

CMLIB subprogram library (LINPACKS sublibrary)

- SSICO** Computes factorization of real symmetric indefinite matrix and estimates its condition. Double precision version is DSICO.
- SSIDI** Uses factorization of real symmetric indefinite matrix to compute its determinant and/or inverse. Double precision version is DSIID.
- SSIFA** Computes factorization of real symmetric indefinite matrix. Double precision version is DSIFA.
- SSISL** Uses factorization of real symmetric indefinite matrix to solve systems. Double precision version is DSIISL.
- SSPCO** Computes factorization of real symmetric indefinite matrix stored in packed form and estimates its condition. Double precision version is DSPCO.
- SSPDI** Uses factorization of real symmetric indefinite matrix stored in packed form to compute its determinant and/or inverse. Double precision version is DSPDI.
- SSPFA** Computes factorization of real symmetric indefinite matrix stored in packed form. Double precision version is DSPFA.
- SSPSL** Uses factorization of real symmetric indefinite matrix stored in packed form to solve systems. Double precision version is DSPSL.

IMSL subprogram library

- LEQ1S** Linear equation solution - indefinite matrix - symmetric storage mode - space economizer solution.
- LEQ2S** Linear equation solution - indefinite matrix - symmetric storage mode - high accuracy solution.

D2b1b : Positive definite**CMLIB subprogram library (LINDRIVES sublibrary)**

- **SPOFS** Factors and solves a symmetric positive definite single precision system of linear equations. Double precision version is DPOFS.
- SPOIR** Factors and solves a symmetric positive definite single precision system of equations and estimates solution accuracy (needs NxN extra storage).

CMLIB subprogram library (LINPACKS sublibrary)

- SCHDC** Compute Cholesky decomposition of real positive definite matrix with optional pivoting. Double precision version is DCHDC.
- SPOCO** Uses Cholesky algorithm to factor real positive definite matrix and estimate its condition. Double precision version is DPOCO.
- SPODI** Uses factorization of real positive definite matrix to compute its determinant and/or inverse. Double precision version is DPODI.
- SPOFA** Uses Cholesky algorithm to factor real positive definite matrix. Double precision version is DPOFA.
- SPOSL** Uses factorization of real positive definite matrix to solve systems. Double precision version is DPOSL.
- SPPCO** Uses Cholesky algorithm to factor real positive definite matrix stored in packed form and estimate its condition. Double precision version is DPPCO.
- SPPDI** Uses factorization of real positive definite matrix stored in packed form to compute its determinant and/or inverse. Double precision version is DPPDI.
- SPPFA** Uses Cholesky algorithm to factor real positive definite matrix stored in packed form. Double precision version is DPPFA.
- SPPSL** Uses factorization of real positive definite matrix stored in packed form to solve systems. Double precision version is DPPSL.

IMSL subprogram library

- LEQT1P** Linear equation solution - positive definite matrix - symmetric storage mode - space economizer solution.
- LEQT2P** Linear equations solution - positive definite matrix - symmetric storage mode - high accuracy solution.
- LINV1P** Inversion of matrix - positive definite symmetric storage mode - space economizer solution.
- LINV2P** Inversion of a matrix - positive definite symmetric storage mode - high accuracy solution.
- LINV3P** In place inverse, equation solution, positive definite matrix - symmetric storage mode.
- LUDECP** Decomposition of a positive definite matrix symmetric storage mode.
- LUELMP** Elimination part of the solution of $Ax=b$ positive definite matrix - symmetric storage mode.
- LUREFP** Refinement of solution to linear equations positive definite matrix - symmetric storage mode.

MATHWARE subprogram library (NASHLIB sublibrary)

- A24CG** Solution of a consistent system of linear equations with symmetric non negative definite coefficient matrix.
- A7CH** Cholesky decomposition of symmetric non-negative definite matrix in compact storage.
- A8CS** Cholesky back substitution for the solution of consistent sets of linear equations with symmetric coefficient matrices, compact storage.
- A9GJ** Bauer Reinsch inversion of a positive definite symmetric matrix by a modification of Gauss-Jordan method.

NAG subprogram library

- **F01ABE** Calculates the accurate inverse of a real symmetric positive definite matrix by Cholesky's method and iterative refinement. (Simplified parameter list). Double precision version is F01ABF.

- F01ACE** Calculates the accurate inverse of a real symmetric positive definite matrix by Cholesky's method and iterative refinement. Double precision version is F01ACF.
- F01ADE** F01ADE calculates the approximate inverse of a real symmetric positive definite matrix by Cholesky's method. Double precision version is F01ADF.
- F01BQE** Forms the Cholesky decomposition of a real symmetric matrix G whose lower triangle only is stored. If G not positive definite, forms Cholesky decomposition of $G+E$, E a diagonal matrix. Double precision version is F01BQF.
- F01BUE** Decomposes a symmetric positive definite matrix into form $U LDL^T U^T$ where U is unit upper triangular, L is unit lower triangular, D is diagonal. Precedes F01BVE. Double precision version is F01BUF.
- F01BXE** Performs the Cholesky factorization $U^T U$ of a real symmetric positive definite matrix A. Double precision version is F01BXF.
- F03AEE** LL^T -factorisation and determinant, real symmetric positive-definite matrix. Double precision version is F03AEF.
- **F04ABE** Simultaneous linear equations (black box), real symmetric positive-definite matrix, accurate solution, multiple right hand sides. Double precision version is F04ABF.
- F04AFE** Calculates the accurate solution of a set of real symmetric positive definite linear equations with multiple right hand sides, $AX = B$, where A has been decomposed into triangular matrices using F03AEE. Double precision version is F04AFF.
- F04AGE** Calculates the approximate solution of a set of real symmetric positive definite linear equations with multiple right hand sides, $AX = B$, where A has been decomposed into triangular matrices using F03AEE. Double precision version is F04AGF.
- F04AQE** Calculates the approximate solution of a set of real symmetric positive definite linear equations with a single right hand side, $Ax = b$, where A has been decomposed into LDLt using F01BQE. (Economical storage.). Double precision version is F04AQF.
- **F04ASE** Simultaneous linear equations (black box), real symmetric positive-definite matrix, accurate solution, one right hand side. Double precision version is F04ASF.
- F04AZE** Calculates the approximate solution of a set of real symmetric positive definite linear equations with a single right hand side, $Ax = b$, where A has been decomposed into triangular matrices using F01BXE. Double precision version is F04AZF.

D2b2 : Positive definite banded

CMLIB subprogram library (LINPACKS sublibrary)

- SPBCO** Uses Cholesky algorithm to compute factorization of real positive definite band matrix and estimates its condition. Double precision version is DPBCO.
- SPBFA** Uses Cholesky algorithm to compute factorization of real positive definite band matrix. Double precision version is DPBFA.
- SPBSL** Uses factorization of real positive definite band matrix to solve systems. Double precision version is DPBSL.

IMSL subprogram library

- LEQ1PB** Linear equation solution - positive definite symmetric band matrix - band symmetric storage mode - space economizer solution.
- LEQ2PB** Linear equation solution - positive definite band symmetric matrix - band symmetric storage mode - high accuracy solution.
- LIN1PB** Inversion of a matrix - positive definite band symmetric matrix - band symmetric storage mode - space economizer solution.
- LIN2PB** Inversion of matrix - positive definite band symmetric matrix - band symmetric storage mode - high accuracy solution.

LUDAPB	Decomposition of a positive definite band symmetric matrix - band symmetric storage mode.
LUELPB	Elimination part of solution of $Ax=b$ positive definite band symmetric matrix band symmetric storage mode.
LUREPB	Refinement of solution to linear equations positive definite band symmetric matrix band symmetric storage mode.

NAG subprogram library

F01MCE	LDLt-factorisation, real symmetric positive-definite variable-bandwidth matrix. Double precision version is F01MCF.
F03AGE	LLt-factorisation and determinant, real symmetric positive-definite band matrix. Double precision version is F03AGF.
• F04ACE	Simultaneous linear equations (black box), real symmetric positive-definite bandmatrix, approximate solution, multiple right hand sides. Double precision version is F04ACF.
F04ALE	Calculates the approximate solution of a set of real symmetric positive definite band linear equations with multiple right hand sides, $AX = B$, where A has been decomposed into triangular matrices using F03AGE. Double precision version is F04ALF.
F04MCE	Calculates the approximate solution of a system of real linear equations with multiple right hand sides, $Ax = B$, where A is a symmetric positive definite variable-bandwidth matrix, which has previously been factorised by F01MCE. Related systems may also be solved. Double precision version is F04MCF.

D2b2a :	Tridiagonal
----------------	-------------

CMLIB subprogram library (LINPACKS sublibrary)

SPTSL	Decomposes real symmetric positive definite tridiagonal matrix and simultaneously solves a system. Double precision version is DPTSL.
--------------	---

D2b4 :	Sparse
---------------	--------

CMLIB subprogram library (YSMP sublibrary)

ODRV	Computes the minimum degree ordering of equations and unknowns for a system of linear algebraic equations in sparse storage mode.
SDRV	Solves sparse symmetric systems of linear algebraic equations by Gaussian elimination without pivoting.

MATHWARE subprogram library (ITPACK sublibrary)

JCG	Iterative solution of large sparse systems of linear equations. Jacobi method, conjugate gradient acceleration, adaptive parameter selection.
JSI	Iterative solution of large sparse systems of linear equations. Jacobi method, Chebyshev acceleration, adaptive parameter selection.
RSCG	Iterative solution of large sparse systems of linear equations. Reduced system method, conjugate gradient acceleration, adaptive.
RSSI	Iterative solution of large sparse systems of linear equations. Reduced system method, Chebyshev acceleration, adaptive.
SOR	Iterative solution of large sparse systems of linear equations. SOR method, adaptive parameter selection.
SSORCG	Iterative solution of large sparse systems of linear equations. SSOR method, conjugate gradient acceleration, adaptive parameter selection.
SSORSI	Iterative solution of large sparse systems of linear equations. SSOR method, Chebyshev acceleration,

adaptive parameter selection.

MATHWARE subprogram library (NASHLIB sublibrary)

A24CG Solution of a consistent system of linear equations with symmetric non negative definite coefficient matrix.

D2c : Complex non-Hermitian matrices

D2c1 : General

CMLIB subprogram library (LINDRIVES sublibrary)

• **CGEFS** Factors and solves a general complex system of linear equations.
CGEIR Factors and solves a general complex system of linear equations and provides estimate of accuracy of the solution (needs NXN extra storage).

CMLIB subprogram library (LINPACKC sublibrary)

CGECO Compute LU factorization of general complex matrix and estimate its condition.
CGEDI Compute determinant and/or inverse of general complex matrix from its LU factors.
CGEFA Compute LU factorization of general complex matrix.
CGESL Use LU factorization of general complex matrix to solve systems.
CQRSL Applies the output of CQRDC to compute coordinate transformations, projections, and least squares solutions (general complex matrix).

IMSL subprogram library

LEQ2C Linear equation solution - complex matrix high accuracy solution.
LEQT1C Matrix decomposition, linear equation solution - space economizer solution complex matrices.

NAG subprogram library

F03AHE LU-factorisation and determinant, complex matrix. Double precision version is F03AHF.
 • **F04ADE** Simultaneous linear equations (black box), complex matrix, approximate solution, multiple right hand sides. Double precision version is F04ADF.
F04AKE Calculates the approximate solution of a set of complex linear equations with multiple right hand sides, $AX = B$, where A has been decomposed into triangular matrices using F03AHE. Double precision version is F04AKF.

PORT subprogram library

CLINQ Solves a complex system of linear equations. Coefficient matrix must be input as two real matrices. Double precision version is DCLINQ.

D2c2 : Banded

CMLIB subprogram library (LINDRIVES sublibrary)

CNBCO Factors a complex band matrix by Gaussian elimination and estimates its condition number.
CNBFA Factors a non-symmetric complex band matrix by elimination.
 • **CNBFS** Factors and solves a general complex band matrix system of linear equations.
CNBIR Factors and solves a general nonsymmetric complex band system of equations and estimates accuracy

of the solution (requires $N \times (2ML + MU)$ extra storage).

CNBSL Solves the nonsymmetric complex band system of equations using factors previously computed.

CMLIB subprogram library (LINPACKC sublibrary)

CGBCO Compute LU factorization of complex band matrix and estimate its condition.

CGBFA Compute LU factorization of general complex band matrix.

CGBSL Uses LU factorization of complex band matrix to solve systems.

D2c2a : Tridiagonal

CMLIB subprogram library (LINPACKC sublibrary)

CGTSL Solves systems with general complex tridiagonal matrix.

D2c3 : Triangular

CMLIB subprogram library (LINPACKC sublibrary)

CTRCO Estimates condition of complex triangular matrix.

CTRDI Computes determinant and/or inverse of complex triangular matrix.

CTRSL Solves systems with complex triangular matrix.

D2c4 : Sparse

D2d : Complex Hermitian matrices

D2d1 : General

D2d1a : Indefinite

CMLIB subprogram library (LINPACKC sublibrary)

CHICO Computes factorization of complex Hermitian indefinite matrix and estimates its condition.

CHIDI Uses factorization of complex Hermitian indefinite matrix to compute its inertia determinant, and/or inverse.

CHIFA Computes factorization of complex Hermitian indefinite matrix.

CHISL Uses factorization of complex Hermitian indefinite matrix to solve systems.

CHPCO Computes factorization of complex Hermitian indefinite matrix stored in packed form and estimates its condition.

CHPDI Uses factorization of complex Hermitian indefinite matrix stored in packed form to compute its inertia, determinant, and inverse.

CHPFA Computes factorization of complex Hermitian indefinite matrix stored in packed form.

CHPSL Uses factorization of complex Hermitian indefinite matrix stored in packed form to solve systems.

CSICO Computes factorization of complex symmetric indefinite matrix and estimates its condition.

CSIDI	Uses factorization of complex symmetric indefinite matrix to compute its determinant and/or inverse.
CSIFA	Computes factorization of complex symmetric indefinite matrix.
CSISL	Uses factorization of complex symmetric indefinite matrix to solve systems.
CSPCO	Computes factorization of complex symmetric indefinite matrix stored in packed form and computes its condition.
CSPDI	Uses factorization of complex symmetric indefinite matrix stored in packed form to compute its determinant and/or inverse.
CSPFA	Computes factorization of complex symmetric indefinite matrix stored in packed form.
CSPSL	Uses factorization of complex symmetric indefinite matrix stored in packed form to solve systems.

D2d1b :	Positive definite
----------------	-------------------

CMLIB subprogram library (LINDRIVES sublibrary)

• CPOFS	Factors and solves positive definite symmetric complex system of linear equations.
CPOIR	Solves positive definite Hermitian complex system of linear equations and estimates the accuracy of the solution (requires NxN extra storage).

CMLIB subprogram library (LINPACKC sublibrary)

CCHDC	Compute Cholesky decomposition of complex positive definite matrix with optional pivoting.
CPOCO	Uses Cholesky algorithm to compute factorization of complex positive definite matrix and estimates its condition.
CPODI	Uses factorization of complex positive definite matrix to compute its determinant and/or inverse.
CPOFA	Uses Cholesky algorithm to compute factorization of complex positive definite matrix.
CPOSL	Uses factorization of complex positive definite matrix to solve systems.
CPPCO	Uses Cholesky algorithm to factor complex positive definite matrix stored in packed form.
CPPDI	Uses factorization of complex positive definite matrix stored in packed form to compute determinant and/or inverse.
CPPFA	Uses Cholesky algorithm to factor complex positive definite matrix stored in packed form.
CPPSL	Uses factorization of complex positive definite matrix stored in packed form to solve systems.

NAG subprogram library

F01BNE	Performs the Cholesky decomposition of a complex positive definite Hermitian matrix, given the lower triangle of the matrix. Double precision version is F01BNF.
F01BPE	Determines the inverse of a complex positive definite Hermitian matrix, given the lower triangle of the matrix. Double precision version is F01BPF.
F04AWE	Calculates the approximate solutions of a set of complex linear equations with multiple right hand sides, $AX = B$, where A is positive definite Hermitian, following the Cholesky decomposition of A by F01BNE. Double precision version is F04AWF.

D2d2 :	Positive definite banded
---------------	--------------------------

CMLIB subprogram library (LINPACKC sublibrary)

CPBCO	Uses Cholesky algorithm to compute factorization of complex positive definite band matrix and estimates its condition.
CPBFA	Uses Cholesky algorithm to compute factorization of complex positive definite band matrix.

CPBSL Uses factorization of complex positive definite band matrix to solve systems.

D2d2a : Tridiagonal

CMLIB subprogram library (LINPACKC sublibrary)

CPTSL Solves systems with complex positive definite tridiagonal matrix.

D2d4 : Sparse

D2e : Associated operations (e.g., matrix reorderings)

D3: Determinants

The determinant of a square matrix is a very useful tool in mathematical theory. However, determinants are very dependent upon the scaling of matrix elements and hence are an unreliable indicator of the numerical difficulties which might arise in trying to solve some problem. Casual users should exercise caution in using determinants in the formulation and solution of problems. Listed below are a number of programs which calculate determinants carefully; they are organized according to matrix type as in class D2.

January 1984

References

1. Dongarra, J. J., J. R. Bunch, C. B. Moler, et al. (1979). *LINPACK Users' Guide*, SIAM, Philadelphia.
2. Forsythe, G. and C. Moler (1967). *Computer Solution of Linear Algebraic Systems*, Prentice-Hall, Englewood Cliffs, New Jersey.

D3 : Determinants

D3a : Real nonsymmetric matrices

D3a1 : General

CMLIB subprogram library (LINPACKS sublibrary)

SGEDI Uses LU factorization of real general matrix to compute its determinant and/or inverse. Double precision version is DGEDI.

IMSL subprogram library

LINV3F In place inverse, equation solution, and/or determinant evaluation - full storage mode.

NAG subprogram library

• **F03AAE** Determinant (black box), real matrix. Double precision version is F03AAF.

F03AFE LU-factorisation and determinant, real matrix. Double precision version is F03AFF.

D3a2 : Banded

CMLIB subprogram library (LINDRIVES sublibrary)

SNBDI Computes the determinant of a single precision band matrix using factors previously computed. Double precision version is **DNBDI**.

CMLIB subprogram library (LINPACKS sublibrary)

SGBDI Uses LU factorization of real band matrix to compute its determinant. (No provision for computing matrix inverse.). Double precision version is **DGBDI**.

D3a2a : Tridiagonal**D3a3 :** Triangular

CMLIB subprogram library (LINPACKS sublibrary)

STRDI Computes determinant and/or inverse of real triangular matrix. Double precision version is **DTRDI**.

D3a4 : Sparse**D3b :** Real symmetric matrices**D3b1 :** General**D3b1a :** Indefinite

CMLIB subprogram library (LINPACKS sublibrary)

SSIDI Uses factorization of real symmetric indefinite matrix to compute its determinant and/or inverse. Double precision version is **DSIDI**.

SSPDI Uses factorization of real symmetric indefinite matrix stored in packed form to compute its determinant and/or inverse. Double precision version is **DSPDI**.

D3b1b : Positive definite

CMLIB subprogram library (LINPACKS sublibrary)

SPODI Uses factorization of real positive definite matrix to compute its determinant and/or inverse. Double precision version is **DPODI**.

SPPDI Uses factorization of real positive definite matrix stored in packed form to compute its determinant and/or inverse. Double precision version is **DPPDI**.

NAG subprogram library

• **F03ABE** Determinant (black box), real symmetric positive-definite matrix. Double precision version is **F03ABF**.

F03AEE LLt-factorisation and determinant, real symmetric positive-definite matrix. Double precision version is **F03AEF**.

D3b2 : Positive definite banded

CMLIB subprogram library (LINPACKS sublibrary)

SPBDI Uses factorization of real positive definite band matrix to compute its determinant. (No provision for matrix inverse.). Double precision version is DPBDI.

NAG subprogram library

• **F03ACE** Determinant (black box), real symmetric positive-definite band matrix. Double precision version is F03ACF.

F03AGE LLt-factorisation and determinant, real symmetric positive-definite band matrix. Double precision version is F03AGF.

D3b2a : Tridiagonal

D3b4 : Sparse

D3c : Complex non-Hermitian matrices

D3c1 : General

CMLIB subprogram library (LINPACKC sublibrary)

CGEDI Compute determinant and/or inverse of general complex matrix from its LU factors.

NAG subprogram library

• **F03ADE** Determinant (black box), complex matrix. Double precision version is F03ADF.

F03AHE LU-factorisation and determinant, complex matrix. Double precision version is F03AHF.

D3c2 : Banded

CMLIB subprogram library (LINDRIVES sublibrary)

CNBDI Computes the determinant of a complex band matrix from previously computed factors.

CMLIB subprogram library (LINPACKC sublibrary)

CGBDI Compute determinant of complex band matrix from its LU factors. (No provision for computing inverse directly.).

D3c2a : Tridiagonal

D3c3 : Triangular

CMLIB subprogram library (LINPACKC sublibrary)

CTRDI Computes determinant and/or inverse of complex triangular matrix.

D3c4 : Sparse

D3d : Complex Hermitian matrices

D3d1 : General

D3d1a : Indefinite

CMLIB subprogram library (LINPACKC sublibrary)

CHIDI Uses factorization of complex Hermitian indefinite matrix to compute its inertia determinant, and/or inverse.

CHPDI Uses factorization of complex Hermitian indefinite matrix stored in packed form to compute its inertia, determinant, and inverse.

CSIDI Uses factorization of complex symmetric indefinite matrix to compute its determinant and/or inverse.

CSPDI Uses factorization of complex symmetric indefinite matrix stored in packed form to compute its determinant and/or inverse.

D3d1b : Positive definite

CMLIB subprogram library (LINPACKC sublibrary)

CPODI Uses factorization of complex positive definite matrix to compute its determinant and/or inverse.

CPPDI Uses factorization of complex positive definite matrix stored in packed form to compute determinant and/or inverse.

NAG subprogram library

F03AME Determinant of a complex Hermitian positive-definite matrix, after factorisation by F01BNE. Double precision version is F03AMF.

D3d2 : Positive definite banded

CMLIB subprogram library (LINPACKC sublibrary)

CPBDI Uses factorization of complex positive definite band matrix to compute determinant. (No provision for computing inverse.).

D3d2a : Tridiagonal

D3d4 : Sparse

D4: Eigenvalues, eigenvectors

The calculation of the eigenvalues and/or eigenvectors of a matrix has received much attention. It was for this class of problems that the first modern mathematical software package, EISPACK, was produced, and most of the routines in this class originated in that project. If you want to compute the eigenvalues and/or eigenvectors for one particular matrix,

you may want to use one of the drivers which invoke a series of EISPACK routines to perform this task. The IMSL and NAG libraries each have such a set of routines. Individual EISPACK routines, as well as a collection of public-domain drivers, called LICEPACK, are available in CMLIB. The EISPACK Guide (see reference below) has a flowchart that is designed to lead one through the maze of routines.

January 1984

References

1. Garbow, B. S., J. M. Boyle, J. J. Dongarra, et al. (1977). *EISPACK Guide Extension*, Springer-Verlag, New York.
2. Parlett, B. N. (1980). *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, New Jersey.
3. Smith, B. T., J. M. Boyle, J. J. Dongarra, et al. (1976). *EISPACK Guide*, 2nd Edition, Springer-Verlag, New York.
4. Wilkinson, J. H. (1965). *The Algebraic Eigenvalue Problem*, Oxford University Press, London.

D4 :	Eigenvalues, eigenvectors
-------------	---------------------------

D4a :	Ordinary eigenvalue problems ($Ax = \lambda x$)
--------------	---

D4a1 :	Real symmetric
---------------	----------------

CMLIB subprogram library (EISPACK sublibrary)

RS Computes eigenvalues and, optionally, eigenvectors of a real symmetric matrix.

RSP Compute eigenvalues and, optionally, eigenvectors of a real symmetric matrix packed into a one dimensional array.

CMLIB subprogram library (LICEPACK sublibrary)

• **SSIEV** Computes the eigenvalues and, optionally, the eigenvectors of a real symmetric matrix.

• **SSPEV** Computes eigenvalues and, optionally eigenvectors of real symmetric matrix stored in packed form.

IMSL subprogram library

EIGRS Eigenvalues and (optionally) eigenvectors of a real symmetric matrix.

MATHWARE subprogram library (NASHLIB sublibrary)

A13ESV Solves eigenproblem for real symmetric matrix via Singular Value Decomposition.

A14JE Jacobi algorithm for the eigenvalues and eigenvectors of a real symmetric matrix.

NAG subprogram library

• **F02AAE** Real symmetric matrix, (black box), all eigenvalues. Double precision version is F02AAF.

• **F02ABE** Real symmetric matrix, (black box), all eigenvalues and eigenvectors. Double precision version is F02ABF.

• **F02BBE** Real symmetric matrix, (black box), selected eigenvalues and eigenvectors. Double precision version is F02BBF.

D4a2 :	Real nonsymmetric
---------------	-------------------

CMLIB subprogram library (EISPACK sublibrary)

RG Computes eigenvalues and, optionally, eigenvectors of a real general matrix.

CMLIB subprogram library (LICEPACK sublibrary)

- **SGEEV** Computes the eigenvalues and, optionally, the eigenvectors of a general real matrix.

IMSL subprogram library

- EIGRF** Eigenvalues and (optionally) eigenvectors of a real general matrix in full storage mode.

NAG subprogram library

- **F02AFE** Real matrix, (black box), all eigenvalues. Double precision version is F02AFF.
- **F02AGE** Real matrix, (black box), all eigenvalues and eigenvectors. Double precision version is F02AGF.
- **F02BCE** Real matrix, (black box), selected eigenvalues and eigenvectors. Double precision version is F02BCF.

PORT subprogram library

- EIGEN** Finds all eigenvalues and eigenvectors of a real matrix. Output consists of pairs of real arrays. Double precision version is DEIGEN.

D4a3 : Complex Hermitian

CMLIB subprogram library (EISPACK sublibrary)

- CH** Computes the eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix.

CMLIB subprogram library (LICEPACK sublibrary)

- **CHIEV** Computes the eigenvalues and, optionally, the eigenvectors of a complex hermitian matrix.

IMSL subprogram library

- EIGCH** Eigenvalues and (optionally) eigenvectors of a complex Hermitian matrix.

NAG subprogram library

- **F02AWE** Complex Hermitian matrix, (black box), all eigenvalues. Double precision version is F02AWF.
- **F02AXE** Complex Hermitian matrix, (black box), all eigenvalues and eigenvectors. Double precision version is F02AXF.

D4a4 : Complex non-Hermitian

CMLIB subprogram library (EISPACK sublibrary)

- CG** Computes the eigenvalues and, optionally, the eigenvectors of a complex general matrix.

CMLIB subprogram library (LICEPACK sublibrary)

- **CGEEV** Computes the eigenvalues and, optionally, the eigenvectors of a general complex matrix.

IMSL subprogram library

- EIGCC** Eigenvalues and (optionally) eigenvectors of a complex general matrix.

NAG subprogram library

- **F02AJE** Complex matrix, (black box), all eigenvalues. Double precision version is F02AJF.
- **F02AKE** Complex matrix, (black box), all eigenvalues and eigenvectors. Double precision version is F02AKF.
- **F02BDE** Complex matrix, (black box), selected eigenvalues and eigenvectors. Double precision version is F02BDF.

D4a5 : Tridiagonal**CMLIB subprogram library (EISPACK sublibrary)**

- BISECT** Compute eigenvalues of symmetric tridiagonal matrix in given interval using Sturm sequencing.
- IMTQL1** Computes eigenvalues of symmetric tridiagonal matrix using implicit QL method.
- IMTQL2** Computes eigenvalues and eigenvectors of symmetric tridiagonal matrix using implicit QL method.
- IMTQLV** Computes eigenvalues of symmetric tridiagonal matrix using implicit QL method. Eigenvectors may be computed later.
- RATQR** Computes largest or smallest eigenvalues of symmetric tridiagonal matrix using rational QR method with Newton correction.
- **RST** Compute eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix.
- RT** Compute eigenvalues and eigenvectors of a special real tridiagonal matrix.
- TQL1** Compute eigenvalue of symmetric tridiagonal matrix by QL method.
- TQL2** Compute eigenvalues and eigenvectors of symmetric tridiagonal matrix.
- TQLRAT** Computes eigenvalues of symmetric tridiagonal matrix using a rational variant of the QL method.
- TRIDIB** Computes eigenvalues of symmetric tridiagonal matrix in given interval using Sturm sequencing.
- TSTURM** Computes eigenvalues of symmetric tridiagonal matrix in given interval and eigenvectors by Sturm sequencing.

IMSL subprogram library

- EQRT1S** Smallest or largest m eigenvalues of a symmetric tridiagonal matrix.
- EQRT2S** Eigenvalues and (optionally) eigenvectors of a symmetric tridiagonal matrix using the QL method.
- EQRT3S** The smallest (or largest) eigenvalues of a tridiagonal matrix in algebraic value whose sum exceeds a given value.

NAG subprogram library

- F02AVE** Real symmetric tridiagonal matrix, all eigenvalues, QL algorithm. Double precision version is F02AVF.
- F02BEE** Real symmetric tridiagonal matrix, selected eigenvalues and eigenvectors, bisection and inverse iteration. Double precision version is F02BEF.
- F02BFE** Real symmetric tridiagonal matrix, selected eigenvalues, bisection. Double precision version is F02BFF.

D4a6 : Banded**CMLIB subprogram library (EISPACK sublibrary)**

- BQR** Computes some of the eigenvalues of a real symmetric band matrix using the QR method with shifts of origin.
- RSB** Computes eigenvalues and, optionally, eigenvectors of real symmetric band matrix.

IMSL subprogram library

- EIGBS** Find some eigenvalues and (optionally) eigenvectors of a real symmetric band matrix.

D4a7 : Sparse**D4b :** Generalized eigenvalue problems (e.g., $Ax = \lambda Bx$)

D4b1 : Real symmetric**CMLIB subprogram library (EISPACK sublibrary)**

- RSG** Computes eigenvalues and, optionally, eigenvectors of real symmetric generalized eigenproblem: $Ax = \lambda Bx$.
- RSGAB** Computes eigenvalues and, optionally, eigenvectors of real symmetric generalized eigenproblem: $ABx = \lambda x$.
- RSGBA** Computes eigenvalues and, optionally, eigenvectors of real symmetric generalized eigenproblem: $BAx = \lambda x$.

IMSL subprogram library

- EIGZS** Eigenvalues and (optionally) eigenvectors of the system $Ax = \lambda Bx$ where A and B are real symmetric matrices and B is positive definite.

MATHWARE subprogram library (NASHLIB sublibrary)

- A15GSE** Solution of the generalized symmetric eigenvalue problem by two applications of the Jacobi algorithm.
- A25RQM** Rayleigh quotient minimization by conjugate gradients.

NAG subprogram library

- **F02ADE** Generalised real symmetric eigenproblem $Ax = kBx$ with positive-definite B (black box), all eigenvalues. Double precision version is F02ADF.
- **F02AEE** Generalised real symmetric eigenproblem $Ax = kBx$ with positive-definite B (black box), all eigenvalues and eigenvectors. Double precision version is F02AEF.

D4b2 : Real general**CMLIB subprogram library (EISPACK sublibrary)**

- RGG** Computes eigenvalues and eigenvectors for real generalized eigenproblem: $Ax = \lambda Bx$.

IMSL subprogram library

- EIGZF** Eigenvalues and (optionally) eigenvectors of the system $Ax = \lambda Bx$ where A and B are real matrices.

MATHWARE subprogram library (NASHLIB sublibrary)

- A10GII** Solves generalized eigenvalue problem by inverse iteration via Gauss elimination.

NAG subprogram library

- **F02BJE** Generalised eigenproblem $Ax = kBx$, QZ algorithm (black box), real matrices, all eigenvalues and (optionally) eigenvectors. Double precision version is F02BJF.

D4b3 : Complex Hermitian**D4b4 : Complex general****IMSL subprogram library**

- EIGZC** Eigenvalues and (optionally) eigenvectors of the system $Ax = \lambda Bx$ where A and B are complex matrices.

NAG subprogram library

- **F02GJE** Generalised eigenproblem $Ax=\lambda Bx$, QZ algorithm (black box), complex matrices, alleigenvalues and (optionally) eigenvectors. Double precision version is F02GJF.

D4b5 : Banded

NAG subprogram library

- F02SDE** Generalised real eigenproblem $Ax=kBx$, where A and B are band matrices, eigenvector by inverse iteration. Double precision version is F02SDF.

D4c : Associated operations

MATHWARE subprogram library (NASHLIB sublibrary)

- A12CVR** Residuals for a complex eigenvalue $e+if$ and eigenvector $x+iy$ of the matrix $A+iZ$.

D4c1 : Transform problem

D4c1a : Balance matrix

CMLIB subprogram library (EISPACK sublibrary)

- BALANC** Balances a general real matrix and isolates eigenvalues whenever possible.
CBAL Balances a complex general matrix and isolates eigenvalues whenever possible.

IMSL subprogram library

- EBALAC** Balance a complex general matrix and isolate eigenvalues whenever possible.
EBALAF Balance a real matrix.

NAG subprogram library

- F01ATE** Balances a real unsymmetric matrix. Double precision version is F01ATF.
F01AVE Balances a complex matrix. Double precision version is F01AVF.

D4c1b : Reduce to compact form

D4c1b1 : Tridiagonal

CMLIB subprogram library (EISPACK sublibrary)

- BANDR** Reduces real symmetric band matrix to symmetric tridiagonal matrix and, optionally, accumulates orthogonal similarity transformations.
HTRID3 Reduces complex Hermitian (packed) matrix to real symmetric tridiagonal matrix by unitary similarity transformations.
HTRIDI Reduces complex Hermitian matrix to real symmetric tridiagonal matrix using unitary similarity transformations.
TRED1 Reduce real symmetric matrix to symmetric tridiagonal matrix using orthogonal similarity transformations.

mations.

TRED2 Reduce real symmetric matrix to symmetric tridiagonal matrix using and accumulating orthogonal transformations.

TRED3 Reduce real symmetric matrix stored in packed form to symmetric tridiagonal matrix using orthogonal transformations.

IMSL subprogram library

EHOUSH Reduction of a complex Hermitian matrix to real symmetric tridiagonal form.

EHOUSS Reduction of a symmetric matrix to symmetric tridiagonal form using a Householder reduction.

NAG subprogram library

F01AGE Gives the Householder reduction of a real symmetric matrix to tridiagonal form for use in F02BEF, F02AVF and F02BFF. Double precision version is F01AGF.

F01AJE Gives the Householder reduction of a real symmetric matrix A to tridiagonal form for use in F02AME. Double precision version is F01AJF.

F01AYE Gives the Householder reduction of a real symmetric matrix A to tridiagonal form for use in F02BEE, F02AVE, and F02BFE. More economical in storage than similar F01AGE. Double precision version is F01AYF.

F01BCE Gives the Householder reduction of a complex Hermitian matrix to tridiagonal form for use in F01AVE or F02AYE. Double precision version is F01BCF.

F01BWE Reduces a symmetric band matrix to tridiagonal form. This routine normally used with F02AVE to find all eigenvalues of A. For selected values, use F02BME. Double precision version is F01BWF.

D4c1b2 : Hessenberg

CMLIB subprogram library (EISPACK sublibrary)

COMHES Reduces complex general matrix to complex upper Hessenberg form using stabilized elementary similarity transformations.

CORTH Reduces complex general matrix to complex upper Hessenberg using unitary similarity transformations.

ELMHES Reduces real general matrix to upper Hessenberg form using stabilized elementary similarity transformations.

ORTHES Reduces real general matrix to upper Hessenberg form using orthogonal similarity transformations.

IMSL subprogram library

EHESSC Reduction of a general complex matrix to complex upper Hessenberg form.

EHESSF Reduction of a nonsymmetric matrix to upper Hessenberg form by orthogonal transformations.

NAG subprogram library

F01AKE Reduces a real unsymmetric matrix to upper Hessenberg form. Double precision version is F01AKF.

F01AME Reduces a complex unsymmetric matrix to complex upper Hessenberg form. Double precision version is F01AMF.

D4c1b3 : Other

CMLIB subprogram library (EISPACK sublibrary)

QZHES The first step of the QZ algorithm for solving generalized matrix eigenproblems. Accepts a pair of real general matrices and reduces one of them to upper Hessenberg form and the other to upper triangular

form using orthogonal transformations. Usually followed by QZIT, QZVAL, QZVEC.

QZIT

The second step of the QZ algorithm for generalized eigenproblems. Accepts an upper Hessenberg and an upper triangular matrix and reduces the former to quasi-triangular form while preserving the form of the latter. Usually preceded by QZHES and followed by QZVAL and QZVEC.

D4c1c : Standardize problem

CMLIB subprogram library (EISPACK sublibrary)

- FIGI** Transforms certain real non-symmetric tridiagonal matrix to symmetric tridiagonal matrix.
- FIGI2** Transforms certain real non-symmetric tridiagonal matrix to symmetric tridiagonal matrix.
- REDUC** Reduces generalized symmetric eigenproblem $Ax = \lambda Bx$, to standard symmetric eigenproblem, using Cholesky factorization.
- REDUC2** Reduces certain generalized symmetric eigenproblems to standard symmetric eigenproblem, using Cholesky factorization.

NAG subprogram library

- F01AEE** Reduces the generalized eigenproblem $Ax = \lambda Bx$, where A is real symmetric and B is real symmetric positive definite, to the standard symmetric eigenproblem. Double precision version is F01AEF.
- F01BDE** Reduces eigenproblems $ABx = \lambda x$, $x^T BA = \lambda x^T$, $BAy = \lambda y$, $y^T AB = \lambda y^T$ to standard symmetric eigenproblem $Qz = \lambda z$. Double precision version is F01BDF.
- F01BVE** Transforms the generalized symmetric eigenproblem $Ax = \lambda Bx$ to equivalent standard eigenproblem $Cy = \lambda y$; A, B, C symmetric band matrices, B positive definite + decomposed $Cy = \lambda y$. Double precision version is F01BVF.

D4c2 : Compute eigenvalues of matrix in compact form

D4c2a : Tridiagonal

CMLIB subprogram library (EISPACK sublibrary)

- BISECT** Compute eigenvalues of symmetric tridiagonal matrix in given interval using Sturm sequencing.
- IMTQL1** Computes eigenvalues of symmetric tridiagonal matrix using implicit QL method.
- IMTQL2** Computes eigenvalues and eigenvectors of symmetric tridiagonal matrix using implicit QL method.
- IMTQLV** Computes eigenvalues of symmetric tridiagonal matrix using implicit QL method. Eigenvectors may be computed later.
- RATQR** Computes largest or smallest eigenvalues of symmetric tridiagonal matrix using rational QR method with Newton correction.
- TQL1** Compute eigenvalue of symmetric tridiagonal matrix by QL method.
- TQL2** Compute eigenvalues and eigenvectors of symmetric tridiagonal matrix.
- TQLRAT** Computes eigenvalues of symmetric tridiagonal matrix using a rational variant of the QL method.
- TRIDIB** Computes eigenvalues of symmetric tridiagonal matrix in given interval using Sturm sequencing.
- TSTURM** Computes eigenvalues of symmetric tridiagonal matrix in given interval and eigenvectors by Sturm sequencing.

NAG subprogram library

- F02AME** Real symmetric matrix, all eigenvalues and eigenvectors, after reduction to tridiagonal form by F01AJE,

QL algorithm. Double precision version is F02AMF.

F02AYE Complex Hermitian matrix, all eigenvalues and eigenvectors, after reduction to real tridiagonal form by F01BCE, QL algorithm. Double precision version is F02AYF.

D4c2b : **Hessenberg**

CMLIB subprogram library (EISPACK sublibrary)

CINVIT Computes eigenvectors of a complex upper Hessenberg matrix associated with specified eigenvalues using inverse iteration.

COMLR Computes eigenvalues of a complex upper Hessenberg matrix using the modified LR method.

COMLR2 Computes eigenvalues and eigenvectors of complex upper Hessenberg matrix using modified LR method.

COMQR Computes eigenvalues of complex upper Hessenberg matrix using the QR method.

COMQR2 Computes eigenvalues and eigenvectors of complex upper Hessenberg matrix.

HQR Computes eigenvalues of a real upper Hessenberg matrix using the QR method.

HQR2 Computes eigenvalues and eigenvectors of real upper Hessenberg matrix using QR method.

INVIT Computes eigenvectors of upper Hessenberg (real) matrix associated with specified eigenvalues by inverse iteration.

NAG subprogram library

F02ANE Complex upper Hessenberg matrix, all eigenvalues, LR algorithm. Double precision version is F02ANF.

F02APE Real upper Hessenberg matrix, all eigenvalues, QR algorithm. Double precision version is F02APF.

F02AQE Real matrix, all eigenvalues and eigenvectors, after reduction to upper Hessenberg form by F01AKE and F01APE, QR algorithm. Double precision version is F02AQF.

F02ARE Complex matrix, all eigenvalues and eigenvectors, after reduction to upper Hessenberg form by F01AME, LR algorithm. Double precision version is F02ARF.

D4c2c : **Other**

CMLIB subprogram library (EISPACK sublibrary)

QZVAL The third step of the QZ algorithm for generalized eigenproblems. Accepts a pair of real matrices, one in quasi-triangular form and the other in upper triangular form and computes the eigenvalues of the associated eigenproblem. Usually preceded by QZHES, QZIT, and followed by QZVEC.

D4c3 : **Form eigenvectors from eigenvalues**

CMLIB subprogram library (EISPACK sublibrary)

BANDV Forms eigenvectors of real symmetric band matrix associated with a set of ordered approximate eigenvalues by inverse iteration.

QZVEC The optional fourth step of the QZ algorithm for generalized eigenproblems. Accepts a matrix in quasi-triangular form and another in upper triangular form and computes the eigenvectors of the triangular problem and transforms them back to the original coordinates. Usually preceded by QZHES, QZIT, QZVAL.

TINVIT Eigenvectors of symmetric tridiagonal matrix corresponding to some specified eigenvalues, using inverse iteration.

NAG subprogram library

F02BKE Compute selected eigenvectors of a real upper Hessenberg matrix by inverse iteration, given estimates of their associated eigenvalues. Double precision version is F02BKF.

F02BLE Compute selected eigenvectors of a complex upper Hessenberg matrix by inverse iteration, given estimates of their associated eigenvalues. Double precision version is F02BLF.

D4c4 : Back transform eigenvectors

CMLIB subprogram library (EISPACK sublibrary)

BAKVEC Forms eigenvectors of certain real non-symmetric tridiagonal matrix from symmetric tridiagonal matrix output from FIGI.

BALBAK Forms eigenvectors of real general matrix from eigenvectors of matrix output from BALANC.

CBABK2 Forms eigenvectors of complex general matrix from eigenvectors of matrix output from CBAL.

COMBAK Forms eigenvectors of complex general matrix from eigenvectors of upper Hessenberg matrix output from COMHES.

CORTB Forms eigenvectors of complex general matrix from eigenvectors of upper Hessenberg matrix output from CORTH.

ELMBAK Forms eigenvectors of real general matrix from eigenvectors of upper Hessenberg matrix output from ELMHES.

ELTRAN Accumulates the stabilized elementary similarity transformations used in the reduction of a real general matrix to upper Hessenberg form by ELMHES.

HTRIB3 Computes eigenvectors of complex Hermitian matrix from eigenvectors of real symmetric tridiagonal matrix output from HTRID3.

HTRIBK Forms eigenvectors of complex Hermitian matrix from eigenvectors of real symmetric tridiagonal matrix output from HTRID1.

ORTBAK Forms eigenvectors of general real matrix from eigenvectors of upper Hessenberg matrix output from ORTHES.

ORTRAN Accumulates orthogonal similarity transformations in reduction of real general matrix by ORTHES.

REBAK Forms eigenvectors of generalized symmetric eigensystem from eigenvectors of derived matrix output from REDUC or REDUC2.

REBAKB Forms eigenvectors of generalized symmetric eigensystem from eigenvectors of derived matrix output from REDUC2.

TRBAK1 Forms the eigenvectors of real symmetric matrix from eigenvectors of symmetric tridiagonal matrix formed by TRED1.

TRBAK3 Forms eigenvectors of real symmetric matrix from the eigenvectors of symmetric tridiagonal matrix formed by TRED3.

NAG subprogram library

F01AFE Derives eigenvectors of several generalized eigenproblems from the corresponding eigenvectors of the related standard symmetric eigenproblem. Double precision version is F01AFF.

F01AHE Derives eigenvectors of a real symmetric matrix from the eigenvectors of the tridiagonal form produced by F01AGE. Double precision version is F01AHF.

F01ALE Transforms eigenvectors of a Hessenberg matrix to those of a real unsymmetric matrix from which the Hessenberg matrix has previously been derived. Double precision version is F01ALF.

F01ANE Transforms eigenvectors of a complex upper Hessenberg matrix to those of a complex unsymmetric matrix from which the Hessenberg matrix has previously been derived. Double precision version is F01ANF.

F01APE Forms the matrix of accumulated transformations from information left by F01AKE. Double precision

version is F01APF.

- F01AUE** Transforms eigenvectors of a balanced matrix to those of the original real unsymmetric matrix. Double precision version is F01AUF.
- F01AWE** Transforms eigenvectors of a balanced matrix to those of the original complex matrix. Double precision version is F01AWF.
- F01AZE** Derives eigenvectors of a real symmetric matrix from eigenvectors of the tridiagonal form produced by F01AYE. Double precision version is F01AZF.
- F01BEE** Derives eigenvectors y of the problems $y^T AB = \lambda y^T$ and $BAy = \lambda y$ from corresponding eigenvectors of derived standard symmetric eigenproblem. Double precision version is F01BEF.

D4c5 : Determine Jordan normal form

D5 : QR decomposition, Gram-Schmidt orthogonalization

CMLIB subprogram library (LINPACKC sublibrary)

- CQRDC** Computes QR decomposition of general complex matrix.

CMLIB subprogram library (LINPACKS sublibrary)

- SQRDC** Computes QR decomposition of real general matrix. Double precision version is DQRDC.

CMLIB subprogram library (SQRLESS sublibrary)

- **SQRANK** For solving linear systems in least squares sense. Computes the QR decomposition of matrix using LINPACK subroutines. Double precision version is DQRANK.

MATHWARE subprogram library (NASHLIB sublibrary)

- A3GR** Given's reduction of a real rectangular matrix.

NAG subprogram library

- F01AXE** Reduces an $m \times n$ real matrix, $m \geq n$, to upper triangular form for use in F04AME and F04ANE. Uses Householder transformations with column pivoting. Double precision version is F01AXF.
- F01QAE** QR-factorisation, real $m \times n$ matrix ($m \geq n$). Double precision version is F01QAF.
- F01QBE** RQ-factorisation, real $m \times n$ matrix ($m \leq n$). Double precision version is F01QBF.
- F02WDE** S.V.D. of a real $m \times n$ matrix, singular values and (optionally) right singular vectors, optionally or conditionally following QR-factorisation ($m \geq n$). Double precision version is F02WDF.
- F05AAE** Schmidt orthogonalisation of n vectors of order m . Double precision version is F05AAF.

D6 : Singular value decomposition

CMLIB subprogram library (EISPACK sublibrary)

- MINFIT** Compute Singular Value Decomposition of rectangular real matrix and solve related Linear Least Squares problem.
- SVD** Compute Singular Value Decomposition of arbitrary real rectangular matrix.

CMLIB subprogram library (LINPACKC sublibrary)

- CSVDC** Computes Singular Value Decomposition of general complex matrix.

CMLIB subprogram library (LINPACKS sublibrary)

SSVDC Computes Singular Value Decomposition of real general matrix. Double precision version is DSVDC.

IMSL subprogram library

LSVDB Singular value decomposition of a bidiagonal matrix.

LSVDF Singular value decomposition of a real matrix.

MATHWARE subprogram library (NASHLIB sublibrary)

A1SVD Singular Value Decomposition by means of orthogonalizing plane rotations.

NAG subprogram library

F01LZE Reduction by similarity transformations, real matrix to bidiagonal form. Double precision version is F01LZF.

F02SZE Singular value decomposition of a real bidiagonal matrix. Double precision version is F02SZF.

F02WAE Singular value decomposition of a real $m \times n$ matrix, singular values and right singular vectors, ($m \geq n$). Double precision version is F02WAF.

F02WBE Singular value decomposition of a real $m \times n$ matrix, singular values and right singular vectors, ($m \leq n$). Double precision version is F02WBF.

F02WCE Singular value decomposition of a real $m \times n$ matrix, singular values and left and right singular vectors. Double precision version is F02WCF.

F02WDE S.V.D. of a real $m \times n$ matrix, singular values and (optionally) right singular vectors, optionally or conditionally following QR-factorisation ($m \geq n$). Double precision version is F02WDF.

D7 : Update matrix decompositions

D7a : LU

D7b : Cholesky

CMLIB subprogram library (LINPACKC sublibrary)

CCHDD Downdates Cholesky factorization of positive definite complex matrix.

CCHEX Updates Cholesky factorization of positive definite complex matrix.

CCHUD Updates Cholesky factorization of positive definite matrix.

CMLIB subprogram library (LINPACKS sublibrary)

SCHDD Downdates Cholesky factorization of real positive definite matrix. Double precision version is DCHDD.

SCHEX Updates Cholesky factorization of real positive definite matrix. Double precision version is DCHEX.

SCHUD Updates Cholesky factorization of real positive definite matrix. Double precision version is DCHUD.

D7c : QR

D7d : Singular value

D8 : Other matrix equations (e.g., $AX+XB=C$)

D9: Overdetermined and underdetermined systems of equations, singular systems, pseudo-inverses

Systems of linear algebraic equations possess either a single (unique) solution, an infinite number of solutions, or no solutions. Those systems possessing a unique solution are called non-singular. A singular system with no solutions is called inconsistent. Consistent singular systems always have an infinite number of solutions. In practical problems overdetermined systems (more rows than columns) are likely to be inconsistent, while underdetermined systems (fewer rows than columns) are likely to possess an infinite number of solutions.

Although there are no solutions to an inconsistent system, it is often useful to find the best approximation to the solution, in some sense. That is, given a system $Ax = b$, we wish to find a vector x which minimizes $\|Ax - b\|$ in some vector norm. The usual case is the L_2 norm, in which case x is a solution in the least squares sense. For consistent systems with an infinite number of solutions one needs some extra condition in order to specify which solution is desired. Often there is some objective function which one wants to minimize over all solutions x . One example of an objective function is $\|x\|$, in which case one determines a "minimum norm solution".

Additional computational difficulty occurs when the problem is "rank deficient". In the case of overdetermined systems this means that the columns of A are linearly dependent, while in the case of underdetermined systems it means that the rows of A are linearly dependent. Rank deficient overdetermined systems occur, for example, in regression models with redundant (or nearly redundant) variables. Some software only applies to the full rank case, and hence one should be careful in selecting software when one suspects rank deficiency.

January 1984

References

1. Lawson, C. L. and R. J. Hanson (1974). *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, New Jersey.

D9 :	Over or underdetermined or singular systems, pseudo-inverses (<i>search also classes D5, D6, K1a, L8a</i>)
------	--

CMLIB subprogram library (EISPACK sublibrary)

MINFIT Compute Singular Value Decomposition of rectangular real matrix and solve related Linear Least Squares problem.

CMLIB subprogram library (FC sublibrary)

BNDACC Introduce new blocks of data for banded least squares problems. See SUBROUTINE BNDSOL.

BNDSOL Solves least squares problem $AX=B$ for banded matrices.

HFTI Solves linear least squares problem $AX=B$.

• **LSEI** Solves linearly constrained least squares problem with equality and inequality constraints. Covariance matrix opt output.

CMLIB subprogram library (LINPACKC sublibrary)

CQRSL Applies the output of CQRDC to compute coordinate transformations, projections, and least squares solutions (general complex matrix).

CMLIB subprogram library (LINPACKS sublibrary)

SQRSL Applies the output of SQRDC to compute coordinate transformations, projections, and least squares solutions (general real matrix). Double precision version is DQRSL.

CMLIB subprogram library (SGLSS sublibrary)

LLSIA Computes least squares solution to $AX=B$ with A an m by n matrix with $m \geq n$ Flexible version of SGLSS.

• **SGLSS** Solves linear least squares problems. Emphasis is put on detecting possible rank deficiency. Performs QR factorization using Householder transformations. Easy-to-use driver for LLSIA and ULSIA.

ULSIA Finds the minimal length solution of the undetermined system of equations $AX=B$ where A is an m by n matrix with $m < n$ Flexible version of SGLSS.

CMLIB subprogram library (SQRLSS sublibrary)

- **SQRLSS** For solving linear systems in least squares sense. Finds solution and residual after matrix factored by SQRANK. Double precision version is DQRLSS.

CMLIB subprogram library (SUDSSODS sublibrary)

- SODS** Solves an overdetermined system of linear equations. For full rank matrices the unique least squares solution is provided. The least squares solution of minimal length can be obtained in the rank deficient case.
- SUDS** Solves underdetermined systems of linear equations. For full rank matrices the minimum norm solution is returned, as well as an orthonormal basis for the null space of the matrix. If the system of equations is inconsistent only the least squares solution of minimal length is computed.

IMSL subprogram library

- LGINF** Generalized inverse of a real matrix.
- LLBQF** Solution of linear least squares problem high accuracy solution.
- LLSQF** Solution of a linear least squares problem.
- OFIMA3** Least squares solution to the matrix equation $AT = B$.

MATHWARE subprogram library (NASHLIB sublibrary)

- A2LSVD** Least squares solution of rectangular linear system by Singular Value Decomposition.

NAG subprogram library

- E02GAE** L1-approximation by general linear function. Double precision version is E02GAF.
- E02GBE** L1-approximation by general linear function subject to linear inequality constraints. Double precision version is E02GBF.
- E02GCE** Calculates an L-infinity solution to an over-determined system of linear equations. Double precision version is E02GCF.
- F01BLE** Calculates the rank and pseudo-inverse of an $m \times n$ real matrix A , $m \geq n$, $\text{rank}(A) = n$, using Householder's factorisation. Double precision version is F01BLF.
- **F04AME** Least-squares, m real equations in n unknowns, $\text{rank} = n$, $m \geq n$, accurate solution (black box). Double precision version is F04AMF.
- F04ANE** Least-squares, m real equations in n unknowns, $\text{rank} = n$, $m \geq n$, approximate solution (after factorisation by F01AXE). Double precision version is F04ANF.
- F04JAE** Least-squares, m real equations in n unknowns, $\text{rank} \leq n$, $m \geq n$, minimal least-squares solution. Double precision version is F04JAF.
- F04JDE** Least-squares, m real equations in n unknowns, $\text{rank} \leq n$, $m \leq n$, minimal least-squares solution. Double precision version is F04JDF.
- F04JGE** Least-squares, m real equations in n unknowns, $\text{rank} \leq n$, $m \geq n$, least-squares solution if $\text{rank} = n$, otherwise minimal least-squares solution. Double precision version is F04JGF.

PORT subprogram library

- CLST2** Finds the least squares solution of a complex linear algebraic system of equations $AX=B$. B may be a matrix. Uses real arithmetic. Double precision version is DCLST2.
- LSTSQ** Finds the least squares solution of a system of linear equations, $AX=B$. B may be a matrix. Double precision version is DLSTSQ.

E: Interpolation

By interpolation we mean the determination of a function which "passes through" given data values. The data may be one-dimensional (curve fitting), or higher dimensional (surface fitting). More generally, interpolation can also refer to requirements that derivative data match (this is sometimes referred to as osculatory interpolation). The purpose of interpolation is to find a functional form which can replace data which is known exactly. This may be design data, for example, in which case one needs a visually pleasing curve which represents the profile of a given object. Interpolation should not be used if the data have noise. In that case approximation techniques such as least squares (see classes K1 or L8) are more appropriate.

January 1984

E : Interpolation

E1: Univariate data (curve fitting)

In this case the problem is to find a function $g(x)$ such that

$$g(x_i) = y_i \quad i = 1, \dots, n$$

where the points (x_i, y_i) are given.

Polynomials have traditionally been used as interpolating functions. Unfortunately, serious computational problems can arise in determining the coefficients of interpolating polynomials. Furthermore polynomial interpolants often wiggle unphysically between data points. Alternative formulations can sometimes help but the more recent trend is to use a piecewise polynomial interpolant, most often a cubic spline.

A **piecewise polynomial** is a function which is composed of different polynomials joined together at points (called knots). A piecewise polynomial function made up of polynomials of degree n is called a **spline** if the function has $n - 1$ continuous derivatives at the knots. Thus, a cubic spline is made up of cubic polynomials joined together so that the entire function has continuous first and second derivatives. Such functions are awkward to deal with analytically because a single neat formula is not available to represent them, but they are perfect for computations where the process of going from x to $g(x)$ can be thought of as a "black box".

When users ask for a spline interpolation to data they normally must input two extra conditions in addition to the data points. That is, the interpolant has two extra degrees of freedom. Often these are used to specify the derivatives of the interpolant at the endpoints, but there are many other choices. Some programs don't have these options which give them the appearance of being easier to use. Rather, the extra degrees of freedom have been fixed internally and are not always consistent with the physical model. With just a little care, however, piecewise polynomials can make excellent interpolants and produce results that usually are more acceptable than polynomials.

The cubic Hermites are another heavily used set of interpolating functions. These functions are also piecewise cubic polynomials like cubic splines, but they are only joined together with one continuous derivative overall. They are easy to compute and are more flexible than splines. For example, if the data are monotonic it is possible to obtain a Hermite cubic which not only interpolates the data but is also monotonic between the data values. Splines often fail to behave this way.

Two common methods used to represent piecewise polynomial functions are the pp and B representations. In the pp representation one simply stores a list of coefficients of the polynomials describing the pp function on each interval. In the B representation the coefficients of an expansion in terms of a particular piecewise polynomial basis (the so-called B-splines) are stored; this is analogous to writing a polynomial as a linear sum of Legendre polynomials. The pp representation requires more storage than the B representation, but is simpler to evaluate. On the other hand, the B-representation is simpler to compute.

Subprograms for piecewise polynomial interpolation often come in pairs. One routine sets up the interpolant from the given data and is called just once. The second evaluates the interpolant or its derivatives at any point by using the results of the first subprogram and may be called many times.

January 1984

References

1. Cox, M. G. (1977). A Survey of Numerical Methods for Data and Function Approximation, in *The State of the Art in Numerical Analysis*, D. Jacobs (Editor), Academic Press, New York, pp. 627-668.

2. de Boor, C. (1978). *A Practical Guide to Splines*, Springer-Verlag, New York.
3. Hayes, J. G. (Editor) (1970). *Numerical Approximation to Functions and Data*, The Athlone Press, London.
4. Schumaker, L. L. (1981). *Spline Functions: Basic Theory*, John Wiley & Sons, New York.

E1 : Univariate data (curve fitting)

E1a : Polynomial splines (piecewise polynomials)

CMLIB subprogram library (BSPLINE sublibrary)

- BINT4** Computes B-spline which interpolates given X,Y data with various end conditions. The "B" representation is used. Double precision version is DBINT4.
- BINTK** Produces B-spline coefficients of k-th order B-spline with given knots and with values at given points. Double precision version is DBINTK.

CMLIB subprogram library (PCHIP sublibrary)

- PCHIC** Determines a piecewise monotone, piecewise cubic Hermite interpolant to given data. User control is available over boundary conditions and/or treatment of points where monotonicity switches direction.
- **PCHIM** Determines a monotone piecewise cubic Hermite interpolant to given data. Default boundary values are provided which are compatible with monotonicity. The interpolant will have an extremum at each point where monotonicity switches direction.
- PCHSP** Determines the cubic spline interpolant to given data. User has control over boundary conditions.

IMSL subprogram library

- **ICSCCU** Cubic spline interpolation (easy-to-use version).
- ICSICU** Interpolatory approximation by cubic splines with arbitrary second derivative end conditions.
- ICSPLN** Cubic spline interpolation with periodic end conditions.
- IQHSCU** Visually pleasing interpolant of one dimensional data via piecewise cubic Hermite function.

NAG subprogram library

- E01BAE** Interpolating functions cubic spline interpolant. Double precision version is E01BAF.
- E02BAE** Least-squares curve fit by cubic splines (including interpolation). Double precision version is E02BAF.

PORT subprogram library

- CSPFI** Fits a cubic spline function to n input data pairs (x,y) with various endpoint conditions. Double precision version is DCSPFI.
- **CSPIN** Interpolates at requested points in given input data using a spline approximation-not a least squares fit. Double precision version is DCSPIN.

E1b : Polynomials

NAG subprogram library

- E01AAE** Interpolated values, one variable, data at unequally spaced points, Aitken's technique. Double precision version is E01AAF.
- E01ABE** Interpolated values, one variable, data at equally spaced points, Everett's formula. Double precision version is E01ABF.

- E01AEE** Interpolating functions, polynomial interpolant, data may include derivative values. Double precision version is E01AEF.
- E02AFE** Least-squares curve fit by polynomials, special data points (including interpolation). Double precision version is E02AFF.

E1c : Other functions (e.g., rational, trigonometric)

NAG subprogram library

- E01RAE** Produces, from a set of function values and corresponding abscissae, the coefficients of an interpolating rational function expressed in continued fraction form. Double precision version is E01RAF.

E2: Multivariate data (surface fitting)

A typical multivariate interpolation problem is to find a function $g(x, y)$ such that

$$g(x_i, y_i) = z_i \quad i = 1, \dots, n$$

where the points (x_i, y_i, z_i) are given. This is a two-dimensional interpolation problem. It corresponds to the determination of a surface passing through a given set of points in three-space. Higher-dimensional problems also occur, although it is difficult to visualize the interpolant in this case.

The difficulty of multidimensional interpolation depends upon the regularity of the data. If data is provided at all points on a rectangular grid, then the problem is relatively easy, since essentially one-dimensional methods may be applied. Most of the software for this case is based upon tensor products of one-dimensional piecewise polynomials. A number of libraries now provide software for two-dimensional gridded data interpolation, and several also handle the three-dimensional case. The more difficult case is when the data is not on a regular grid; this is the so-called scattered data interpolation problem. One of the difficulties here is that data may be dense in some areas and very sparse (or non-existent) in others; no numerical method can be expected to reproduce the underlying function in the absence of data. One should avoid using software for this problem without carefully examining the result. Much less software is available for this case, and what is available is restricted to two dimensions.

Numerical methods for scattered data interpolation can be classified as either global or local, although several hybrids have also been proposed. In a global method one writes the interpolant as a linear combination of functions each of which is non-zero at a single interpolation point and zero at all others. Such interpolants are easy to construct, expensive to evaluate, and sometimes lead to visually displeasing surfaces. Local methods often begin by triangularizing the data (determining a triangular grid with vertices corresponding to interpolation points). A separate polynomial function is then determined on each triangle which interpolates the local data. Although the resulting interpolant is easy to evaluate, determining it is more complex, especially if continuity restrictions are imposed.

January 1984

References

1. Barnhill, R. E. (1977). Representation and Approximation of Surfaces, in *Mathematical Software III*, J. R. Rice (Editor), Academic Press, New York, pp. 69-120.
2. Franke, R. (1982). Scattered Data Interpolation: A Comparison of Methods, *Mathematics of Computation*, Vol. 38, pp 181-200.
3. Lawson, C. L. (1977). Software for C^1 Surface Interpolation, in *Mathematical Software III*, J. R. Rice (Editor), Academic Press, New York, pp. 161-194.
4. Prenter, P. M. (1975). *Splines and Variational Methods*, John Wiley and Sons, New York, Chapter 6.

E2 : Multivariate data (surface fitting)

E2a : Gridded

- B2INK** Computes parameters of a piecewise-polynomial that interpolates a given set of two-dimensional gridded data. (Use B2VAL to evaluate function.). Double precision version is DB2INK.
- B3INK** Computes parameters of a piecewise-polynomial that interpolates a given set of three-dimensional gridded data. (Use B3VAL to evaluate function.). Double precision version is DB3INK.

IMSL subprogram library

- IBCCCU** Bicubic spline two-dimensional coefficient calculator.
- IBCIEU** Bicubic spline two-dimensional interpolator.

NAG subprogram library

- E01ACE** Interpolated values, two variables, data on rectangular grid, fitting bicubic spline. Double precision version is E01ACF.

E2b : Scattered

CMLIB subprogram library (LOTPS sublibrary)

- LOTPS** Passes smooth function thru points (XI(K),YI(K),FI(K),I=1..NPI) and returns an array of interpolated values on user specified grid.

IMSL subprogram library

- IQHSCV** Smooth surface fitting with irregularly distributed data points.

E3 : Service routines (e.g., grid generation, evaluation of fitted functions) (*search also class N5*)

CMLIB subprogram library (BSPLINE sublibrary)

- BFQAD** Integrates function times derivative of B-spline from X1 to X2. The B-spline is in "B" representation. Double precision version is DBFQAD.
- BSPDR** Constructs divided difference table from "B" representation of B-spline for a derivative calculation. Double precision version is DBSPDR.
- BSPEV** Calculates the value of a spline and its derivatives at X from its "B" representation. Double precision version is DBSPEV.
- BSPPP** Converts from "B" representation of B-spline to piecewise polynomial representation. Double precision version is DBSPPP.
- BSPVD** Calculates value and derivatives of order less than NDERIV of all B-spline basis functions which do not vanish at X. Double precision version is DBSPVD.
- BSPVN** Calculates the value of all (possibly) nonzero B-spline basis functions at X of a given order. Double precision version is DBSPVN.
- BSQAD** Computes the integral of a B-spline from X1 to X2. The B-spline must be in "B" representation. Double precision version is DBSQAD.
- BVALU** Calculates (at X) the value of the IDERIV-th derivative of the B-spline from its "B" representation. Double precision version is DBVALU.
- INTRV** Computes the index into a knot or breakpoint sequence corresponding to a given point X. Double precision version is DINTRV.
- PFQAD** Computes integral on (X1,X2) of product of function and the ID-th derivative of B-spline which is in piecewise polynomial representation. Double precision version is DPFQAD.
- PPQAD** Computes the integral of a B-spline from X1 to X2. The B-spline must be in piecewise polynomial representation. Double precision version is DPPQAD.

PPVAL Calculates (at X) the value of the IDERIV-th derivative of the B-spline from its piecewise polynomial representation. Double precision version is DPPVAL.

CMLIB subprogram library (PCHIP sublibrary)

CHFDV Evaluates a cubic polynomial and its first derivative at an array of points. The polynomial must be given in Hermite form.

CHFEV Evaluates a cubic polynomial given in Hermite form at an array of points.

PCHFD Evaluates a piecewise cubic Hermite function and its first derivative at an array of points.

PCHFE Evaluates a piecewise cubic Hermite function at an array of points.

PCHIA Evaluates the definite integral of a piecewise cubic Hermite function over an arbitrary interval.

PCHID Evaluates the definite integral of a piecewise cubic Hermite function over an interval whose endpoints are data points.

PCHMC Checks a cubic Hermite function for monotonicity.

CMLIB subprogram library (TENSORBS sublibrary)

B2VAL Evaluates the two-dimensional interpolating function computed by B2INK or one of its partial derivatives. Double precision version is DB2VAL.

B3VAL Evaluates the three-dimensional interpolating function computed by B3INK or one of its partial derivatives. Double precision version is DB3VAL.

IMSL subprogram library

DBCEVL Bicubic spline mixed partial derivative evaluator.

DCSEVU Cubic spline first and second derivative evaluator.

DCSQDU Cubic spline quadrature.

IBCEVL Evaluation of a bicubic spline.

ICSEVU Evaluation of a cubic spline.

NAG subprogram library

E01RBE Evaluates continued fractions of the form produced by NAG library routine E01RAE. Double precision version is E01RBF.

E02AEE Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form (simplified parameter list). Double precision version is E02AEF.

E02AHE Derivative of fitted polynomial in Chebyshev series form. Double precision version is E02AHF.

E02AKE Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form. Double precision version is E02AKF.

E02BBE Evaluation of fitted functions, cubic spline as E02BAE, function only. Double precision version is E02BBF.

E02BCE Evaluation of fitted functions, cubic spline as E02BAE, function and derivatives. Double precision version is E02BCF.

E02BDE Evaluation of fitted functions, cubic spline as E02BAE, definite integral. Double precision version is E02BDF.

E02CBE Evaluation of fitted functions, polynomial in two variables as E02CAE. Double precision version is E02CBF.

E02DBE Evaluation of fitted functions, bicubic spline as E02DAE. Double precision version is E02DBF.

E02ZAE Sort 2-d data into panels for fitting or evaluating bicubic splines. Double precision version is E02ZAF.

PORT subprogram library

BSPL1 Evaluates, at a given set of points in a specified mesh interval, basis splines together with selected orders of derivatives. Double precision version is DBSPL1.

BSPLD	Evaluates at a given set of points in a specified mesh interval, basis splines and their derivatives. Double precision version is DBSPLD.
BSPLI	Obtains the integrals of basis splines, from the left-most mesh point to a specified set of points. Double precision version is DBSPLI.
BSPLN	Evaluates at a given set of points in a specified mesh interval, all the basis splines which are nonzero in that interval. Double precision version is DBSPLN.
CSPFE	Evaluates a cubic spline function which has already been fit to n input data pairs (x,y) by CSPFI. Double precision version is DCSPFE.
DLUMD	Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points.
EEBSF	Estimates the error in a given B-spline fit to a function, f , by refining the mesh. Double precision version is DEEBSF.
EEBSI	Estimates the error in a given B-spline fit to a function f by refining the mesh intervals selected by user. Double precision version is DEEBSI.
EESFF	Finds the maximum absolute error in a given B-spline fit to a function, f . Double precision version is DEESFF.
EESFI	Finds the maximum absolute error in a given B-spline fit to a function, f , on a set of user selected intervals. Double precision version is DEESFI.
IDMNPB	Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each mesh interval.
ILUMB	Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points for B-spline use. Double precision version is IDLUMB.
ILUMD	Given a basic mesh, this subdivides each interval into the same number or uniformly spaced points for B-spline use. Double precision version is IDLUMD.
IMNPB	Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each mesh interval.
IPUMB	Given a basic mesh, this subdivides each interval. Number of points per interval can vary, but uniform in each subdivision. Double precision version is IDPUMB.
IPUMD	Given a basic mesh, this subdivides each interval with a variable number of points. Points are uniform in each interval. Double precision version is IDPUMD.
IUMB	Given interval endpoints, this generates a uniform mesh for B-spline use. Double precision version is IDUMB.
IUMD	Given interval endpoints, this generates a uniform mesh. Double precision version is IDUMD.
LUMB	Given a basic mesh, this subdivides each interval uniformly for B-spline use. Multiplicities are allowed. Double precision version is DLUMB.
LUMD	Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points. Double precision version is IDUMD.
MNPB	Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each interval. Double precision version is DMNPB.
PUMB	Given a basic mesh, this subdivides each interval into a uniform but variable number of points. Multiplicities can occur. Double precision version is DPUMB.
PUMD	Given a basic mesh, this subdivides each interval into a uniform but variable number of points. Double precision version is DPUMD.
SPLN1	Evaluates a function and derivatives described previously by an expansion in terms of B-splines. Double precision version is DSPLN1.
SPLN2	Evaluates a function described by a previously determined expansion in B-splines. More flexible than SPLN1. Double precision version is DSPLN2.
SPLND	Evaluates at a given set of points a function described by a previously determined expansion in terms of B-splines. Double precision version is DSPLND.
SPLNE	Evaluates at a set of points, a function described by a previously determined expansion in terms of

B-splines. Double precision version is DSPLNE.

- SPLNI** Integrates a function described previously by an expansion in terms of B-splines. Several integrations can be performed in one call. Double precision version is DSPLNI.
- UMB** Given interval endpoints, this generates a uniform mesh, with needed multiplicities for B-spline use. Double precision version is DUMB.
- UMD** Given interval endpoints, this generates a uniform mesh of distinct points. Double precision version is DUMD.

F: Solution of Nonlinear Equations

This chapter covers the solution of a system of m nonlinear equations for m unknowns. Most routines for this problem find a single solution of the given system based upon a starting guess provided by the user. Convergence cannot usually be guaranteed unless the starting guess is reasonably good. A special case is $m = 1$, a single nonlinear equation. Globally convergent algorithms are available for this case, provided one knows an interval in which the solution lies. Algorithms for the case of a single polynomial equations are even more specialized, and a number of programs are available which reliably compute all the zeros of a polynomial of moderate order.

January 1984

F :	Solution of nonlinear equations
------------	---------------------------------

F1: Single equation

Solving one nonlinear equation,

$$f(x) = 0,$$

for one root is easy if the user has a good estimate of the answer or can bound the interval where the root is to be found.

Usually, the user provides a guess at the root, a Fortran SUBROUTINE or FUNCTION to calculate f for any given x , and a specification of the accuracy desired. Some programs stop when $|f(x)|$ is small enough, and others stop when x is believed to be close to a root. Either tolerance might be absolute or relative, depending on the program. The most reliable programs require the user to provide an interval (a, b) in which a root lies. By evaluating f at a sequence of points in the interval, ever smaller sub-intervals are produced which contain the root. Some such programs require $f(a) \geq 0$ and $f(b) \leq 0$ or the reverse, and then can guarantee finding a root. Programs which use derivatives of f , or estimates of the derivatives, are less reliable, but may be somewhat faster than the programs which shrink the interval surrounding the root. Polynomial equations should ordinarily be solved by a special purpose program.

In Fortran subprograms that allow the user to specify a name for the FUNCTION subprogram which evaluates the function to be "zeroed" the user should be sure that this name appears

- (1) in an EXTERNAL statement in the main program,
- (2) in the subroutine CALL, and
- (3) as the name of a FUNCTION.

January 1984

References

1. Shampine, L. F. and R. C. Allen Jr. (1973). *Numerical Computing: An Introduction*, Saunders, Philadelphia.

F1 :	Single equation
-------------	-----------------

F1a :	Smooth
--------------	--------

F1a1: Polynomial

Programs for solving polynomial equations,

$$a_0 + a_1x + a_2x^2 + \dots + a_nx^n = 0,$$

are quite reliable. They find all n roots of the polynomial. The difficulty is that high-degree polynomials are inherently unstable. Small errors in calculating the a 's can give large errors in the roots. Fortunately, many problems which require solving a polynomial equation, have formulations which are preferable. The casual user should consult an expert before attempting to solve such a problem with n larger than five.

January 1984

References

1. Jenkins, M. A. and J. F. Traub (1970). A Three-Stage Variable-Shift Iteration for Polynomial Zeros and Its Relation to the Generalized Rayleigh Iteration, *Numerische Mathematik*, Vol. 14, No. 3, pp. 253-263.

2. Jenkins, M. A. and J. F. Traub (1975). Principles for Testing Polynomial Zerofinding Programs, *ACM Transactions On Mathematical Software*, Vol. 1, No. 1, pp. 28-34.

F1a1 : Polynomial

F1a1a : Real coefficients

CMLIB subprogram library (CPQR79 sublibrary)

RPQR79 Computes all the zeros of a general real polynomial using eigenvalue methods, requiring NxN storage for Nth degree polynomial.

CMLIB subprogram library (CPZERO sublibrary)

RPZERO Computes all the zeros of a polynomial with real coefficients. Error bounds are also computed. Uses Newton's Method for systems.

IMSL subprogram library

ZPOLR Zeros of a polynomial with real coefficients (Laguerre).

ZQADR Zeros of a quadratic with real coefficients.

ZRPOLY Zeros of a polynomial with real coefficients (Jenkins-Traub).

NAG subprogram library

C02AEE All zeros of polynomial, Grant & Hitchin's method, real coefficients. Double precision version is C02AEF.

PORT subprogram library

RPOLY Finds zeros of a polynomial with real coefficients. Output zeros are in a pair of arrays, for real and imaginary part.

F1a1b : Complex coefficients

CMLIB subprogram library (CPQR79 sublibrary)

CPQR79 Computes all the zeros of a general complex polynomial using eigenvalue methods, requiring NxN storage for Nth degree polynomial.

CMLIB subprogram library (CPZERO sublibrary)

CPZERO Computes all the zeros of a polynomial with complex coefficients. Error bounds are also obtained. Uses Newton's Method for systems.

IMSL subprogram library

ZCPOLY Zeros of a polynomial with complex coefficients (Jenkins-Traub).

ZQADC Zeros of a quadratic with complex coefficients.

NAG subprogram library

C02ADE All zeros of polynomial, Grant & Hitchin's method, complex coefficients. Double precision version is C02ADF.

PORT subprogram library

CPOLY Finds the zeros of a polynomial with complex coefficients. Uses two real arrays to represent complex

numbers, inconvenient. Double precision version is DCPOLY.

F1a2 : Nonpolynomial

IMSL subprogram library

ZANLYT Zeros of an analytic complex function using the Muller method with deflation.
ZFALSE Zero of a function given an interval containing the zero.
ZREAL1 The real zeros of a real function - to be used when initial guesses are poor.
ZREAL2 The real zeros of a real function - to be used when initial guesses are good.

NAG subprogram library

C05AJE Zero of continuous function of one variable, from a given starting value, continuation method. Double precision version is C05AJF.
C05AXE Zero of continuous function of one variable, from a given starting value, continuation method (reverse communication). Double precision version is C05AXF.

F1b : General (no smoothness assumed)

CMLIB subprogram library (ZEROIN sublibrary)

ZEROIN Finds a zero of a user defined function on an interval given the endpoints A and B such that $F(A) \cdot F(B) < 0$.

IMSL subprogram library

ZBRENT Zero of a function which changes sign in a given interval (Brent algorithm).

MATHWARE subprogram library (NASHLIB sublibrary)

A18RF Root finding or minimization by bisection and false position.

NAG subprogram library

C05ADE Zero of continuous function of one variable, in a given interval, Bus & Dekker algorithm. Double precision version is C05ADF.
C05AGE Zero of continuous function of one variable, from a given starting value, search for interval, Bus & Dekker algorithm. Double precision version is C05AGF.
C05AVE Zero of continuous function of one variable, search for interval containing zero (reverse communication). Double precision version is C05AVF.
C05AZE Zero of continuous function of one variable, in a given interval, Bus & Dekker algorithm (reverse communication). Double precision version is C05AZF.

PORT subprogram library

ZERO Finds a single real root of a function within an interval specified by the user. Double precision version is DZERO.

F2: System of equations

Solving a system of nonlinear equations,

$$F(x) = 0,$$

where x is a vector and F is a vector-valued function, both of length n , is not easy unless the user can provide a good estimate of the root.

Usually the user provides this estimate, a Fortran SUBROUTINE subprogram to calculate F for any given x , and a specification of the accuracy required. Some programs stop when $\|F(x)\|$ is small enough, and others stop when x is believed to be close to a root. Some programs interpret $\|F(x)\|$ as the largest component of F , others as the square root of the sum of squares of the components. Tolerances might be absolute or relative, depending on the program.

Programs to solve systems of nonlinear equations usually approximate numerically the "Jacobian matrix", a matrix of partial derivatives of the components of F with respect to the components of x . Some programs allow the user to provide this matrix, and therefore can be more reliable. The Jacobian matrix is used in a matrix version of Newton's method. A sequence of x 's is constructed which may converge to a root. Programs vary as to when they quit if the process does not seem to be converging.

In Fortran subprograms that allow the user to specify a name for the FUNCTION subprogram which evaluates the function to be "zeroed" the user should be sure that this name appears

- (1) in an EXTERNAL statement in the main program,
- (2) in the subroutine CALL, and
- (3) as the name of a FUNCTION.

January 1984

References

1. Allgower, E. and K. Georg (1980). Simplicial and Continuation Methods for Approximating Fixed Points and Solutions to Systems of Equations, *SIAM Review*, Vol. 22, No. 1, pp. 28-85.
2. Dennis, J. E. and R. B. Schnabel (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ.
3. More, J.J. and M.Y. Cosnard (1979). Numerical Solution of Nonlinear Equations, *ACM Transactions on Mathematical Software*, Vol. 5, No. 1, pp. 64-85.
4. Ortega, J. M. and W. C. Rheinboldt (1970). *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York.
5. Rheinboldt, W. C. (1974). *Methods for Solving Systems of Nonlinear Equations*, SIAM, Philadelphia.
6. Sargent, R. W. (1981). A Review of Methods for Solving Nonlinear Algebraic Equations, in *Foundations of Computer Aided Chemical Process Design*, R. S. H. Mah and W. D. Seider (Editors), Vol. 1, Engineering Foundation, New York.

F2 :	System of equations
-------------	---------------------

F2a :	Smooth
--------------	--------

CMLIB subprogram library (SNLS1E sublibrary)

- SNSQ** Finds a zero of a system of N nonlinear equations in N variables by a modification of the Powell hybrid method. Flexible usage.
- **SNSQE** Finds a zero of a system of N nonlinear equations in N variables by a modification of Powell's hybrid method. An easy to use driver for SNSQ.
- **SOS** Finds a zero of a system of N nonlinear equations in N unknowns using Brown's method.

IMSL subprogram library

- ZSCNT** Solve a system of nonlinear equations.
- ZSPOW** Solve a system of nonlinear equations (uses function values only).

NAG subprogram library

- **C05NBE** Easy-to-use routine to find a zero of a system of N nonlinear functions in N variables by a modification of the Powell hybrid method. Derivatives of the function are not required. Double precision version is C05NBF.
- C05NCE** Finds a zero of a system of N nonlinear functions in N variables by a modification of Powell's hybrid method. Derivatives of the functions are not required. (Comprehensive version of C05NBE.). Double

precision version is C05NCF.

- **C05PBE** Easy-to-use routine to find a zero of a system of N nonlinear functions in N variables by a modification of Powell's hybrid method. The user must provide the Jacobian. Double precision version is C05PBF.
- C05PCE** Finds a zero of a system of N nonlinear functions in N variables by a modification of Powell's hybrid method. The user must provide the Jacobian. (Comprehensive version of C05PBE.). Double precision version is C05PCF.

PORT subprogram library

- **ZONE** Finds a solution of a system of non-linear equations. Double precision version is DZONE.
- ZONEJ** Finds a solution of a system of non-linear equations. User must provide a SUBROUTINE to compute the Jacobian matrix. Double precision version is DZONEJ.

F2b : General (no smoothness assumed)

F3 : Service routines (e.g., check user- supplied derivatives)

IMSL subprogram library

ZSRCH Generate points in an n dimensional space.

NAG subprogram library

- C05ZAE** Checks the user-provided Jacobian prior to use in C05PBE or C05PCE. Double precision version is C05ZAF.
- E04HCE** Check user's routine calculating first derivatives of function. Double precision version is E04HCF.
- E04HDE** Check user's routine calculating second derivatives of function. Double precision version is E04HDF.

G: Optimization

The general problem addressed in this chapter is that of minimizing or maximizing a function $f(x)$ of one or more variables $x_T = (x_1, x_2, \dots, x_n)$. In addition, the values of x which are allowed might be required to satisfy an expression such as $c(x) = 0$, or $d(x) \geq 0$, or that the x_i can assume only integer values. Such expressions are referred to as constraints, and a number of constraints are referred to collectively as the constraint set C (e.g. $C = \{c_i(x) \geq 0, x_j \geq 0, \text{ for } i = 1, 2, \dots, m; j = 1, 2, \dots, n\}$).

Alternative solution methods are chosen depending upon the functional form of f and the constraint set C . For example, if f and each constraint of C is linear, then linear programming techniques can be applied. Problems with tens of thousands of variables and thousands of constraints can be solved using such techniques. If, in addition, the constraint set can be formulated as a network (arcs representing variables and nodes representing constraints) then network algorithms can be used to solve problems having millions of arcs and thousands of nodes.

Once the problem becomes nonlinear, that is, either the objective function or one or more of the constraints are nonlinear, then computational effort increases and choosing among algorithms becomes more difficult. Different algorithms are designed for solving various classes of nonlinear programming problems, such as unconstrained optimization problems, problems with inequality constraints, problems with equality constraints, and problems with both types of constraints. Within each of these categories, different algorithms make specific assumptions about the problem structure. For example, in unconstrained optimization, some procedures assume that the objective function is differentiable and use gradient values, whereas other algorithms do not make this assumption and rely primarily on function evaluations. (Algorithms requiring derivatives often provide the option of finite difference derivatives.) For problems with equality constraints, some algorithms can only handle linear constraints, while others can handle nonlinear constraints as well.

In general, one should use algorithms which exploit the structure of the problem. Thus, one should use a network code over a general linear programming code when feasible, especially for problems having all integer data. Problems having only linear constraints should be solved by an algorithm which fully exploits this structure. Problems for which one can provide derivatives should be solved by algorithms that capitalize on this. Similarly, it is much more efficient to use an algorithm that exploits least-squares structure when it exists. Other types of problems for which specific techniques have been developed include least absolute value, quadratic objective function with linear constraints, geometric programming problems, linear-fractional objective function with linear constraints, fixed point problems, shortest path, longest path, and minimum spanning tree problems, and a variety of integer programming problems where the linear constraint set has a specific structure.

Finally, inherent structure in the underlying application can sometimes be exploited to create a more tractable mathematical model. Nevertheless, there are a variety of special-purpose algorithms that can be constructed for giving good, fast, approximate solutions with known error bounds for some intractable problems.

January 1984

References

1. Bazaraa, M. S. and C. M. Shetty (1979). *Nonlinear Programming*, John Wiley & Sons, New York.
2. Dennis, J. E. and R. B. Schnabel (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ.
3. Gill, P. E., et al. (1979). The Design and Structure of a Fortran Program Library for Optimization, *ACM Transactions on Mathematical Software*, Vol. 5., No. 3, pp. 259-283.
4. Gill, P. E., W. Murray, and M. H. Wright (1981). *Practical Optimization*, Academic Press, New York.
5. Hiebert, K. L. (1981). An Evaluation of Mathematical Software that Solves Nonlinear Least Squares Problems, *ACM Transactions on Mathematical Software*, Vol. 7, No. 1, pp. 1-16.
6. Hillier, F. S. and G. J. Lieberman (1980). *Introduction to Operations Research*, Holden-Day, San Francisco.
7. Hu, T. C. (1982). *Combinatorial Algorithms*, Addison-Wesley, Menlo Park.
8. Wagner, H. M. (1975). *Principles of Management Science*, Prentice-Hall, Englewood Cliffs, NJ.

G : Optimization (search also classes K, L8)

G1 : Unconstrained

G1a : Univariate

G1a1 : Smooth function

G1a1a : User provides no derivatives

IMSL subprogram library

ZXLSF One-dimensional minimization of a smooth function using safeguarded quadratic interpolation.

NAG subprogram library

E04ABE Minimum, function of one variable using function values only. Double precision version is E04ABF.

G1a1b : User provides first derivatives

NAG subprogram library

E04BBE Minimum, function of one variable, using first derivative. Double precision version is E04BBF.

G1a1c : User provides first and second derivatives

G1a2 : General function (no smoothness assumed)

IMSL subprogram library

ZXGSN One-dimensional unimodal function minimization using the golden section search method.

ZXGSP One-dimensional unimodal function minimization using the golden section search method - data parameters specified.

MATHWARE subprogram library (NASHLIB sublibrary)

A16GS Grid or equal interval search along a line.

A17LS Success failure linear search with parabolic inverse interpolation.

A18RF Root finding or minimization by bisection and false position.

PORT subprogram library

FMIN Finds an approximate local minimum of a univariate user defined EXTERNAL function, f. Double precision version is DFMIN.

G1b : Multivariate

G1b1 : Smooth function

G1b1a : User provides no derivatives

CMLIB subprogram library (NL2SN sublibrary)

SMSNO Minimize a general unconstrained objective function using finite difference gradients and secant Hessian approximations. Double precision version is DSMSNO.

IMSL subprogram library

- ZXCGR** A conjugate gradient algorithm for finding the minimum of a function of n variables.
ZXMIN Minimum of a function of n variables using a quasi-Newton method.

NAG subprogram library

- **E04CGE** Unconstrained minimum, function of several variables (easy-to-use), using function values only, quasi-Newton algorithm. Double precision version is E04CGF.
E04JBE Minimum, function of several variables, simple bounds (comprehensive), using function values only, quasi-Newton algorithm. Double precision version is E04JBF.

G1b1b : User provides first derivatives

CMLIB subprogram library (NL2SN sublibrary)

- SUMSL** Minimizes a general unconstrained objective function using analytic gradient and a Hessian approximation from a secant update. Double precision version is DSUMSL.

MATHWARE subprogram library (NASHLIB sublibrary)

- A21VM** Variable metric minimization method.
A22CGM Function minimization by conjugate gradients.

NAG subprogram library

- E04DBE** Unconstrained minimum, function of several variables (comprehensive), using first derivatives, conjugate direction algorithm. Double precision version is E04DBF.
 • **E04DEE** Unconstrained minimum, function of several variables (easy-to-use), using first derivatives, quasi-Newton algorithm. Double precision version is E04DEF.
 • **E04DFE** Unconstrained minimum, function of several variables (easy-to-use), using first derivatives, modified Newton algorithm. Double precision version is E04DFF.

G1b1c : User provides first and second derivatives

CMLIB subprogram library (NL2SN sublibrary)

- HUMSL** Minimizes a general unconstrained objective function using (analytic) gradient and Hessian provided by the user. Double precision version is DHUMSL.

NAG subprogram library

- **E04EBE** Unconstrained minimum, function of several variables (easy-to-use), using first and second derivatives, modified Newton algorithm. Double precision version is E04EBF.

G1b2 : General function (no smoothness assumed)

MATHWARE subprogram library (NASHLIB sublibrary)

- A19NM** Nelder Mead simplex function minimization. A short routine, but often quite successful.

NAG subprogram library

- E04CCE** Unconstrained minimum, function of several variables (comprehensive), using function values only, simplex algorithm. Double precision version is E04CCF.

G2 :	Constrained
-------------	-------------

G2a :	Linear programming
--------------	--------------------

G2a1 :	Dense matrix of constraints
---------------	-----------------------------

IMSL subprogram library

- ZX0LP** Solve the linear programming problem (phase one or phase two) via the revised simplex algorithm.
- **ZX3LP** Solve the linear programming problem via the revised simplex algorithm easy to use version.
- **ZX4LP** Solve the linear programming problem via the revised simplex algorithm (alternate easy to use version).

NAG subprogram library

- H01ABE** Linear programming, simplex algorithm, one iteration. Double precision version is H01ABF.
- H01ADE** Linear programming, revised simplex method. Double precision version is H01ADF.
- H01BAE** Linear programming, numerically stable form of simplex method. Double precision version is H01BAF.

G2a2 :	Sparse matrix of constraints
---------------	------------------------------

CMLIB subprogram library (SPLP sublibrary)

- SPLP** Solves linear optimization problems, that is, it minimizes the linear function $(\text{COSTS})^T x$ subject to $Ax=w$, where the entries of the vectors x and w may have simple upper or lower bounds. Uses a sparse storage mode for the matrix A and out-of-core scratch storage.

G2b :	Transportation and assignments problem
--------------	--

NAG subprogram library

- H03ABE** Solves the classical Transportation ("Hitchcock") problem. Double precision version is H03ABF.

G2c :	Integer programming
--------------	---------------------

G2c1 :	Zero/one
---------------	----------

G2c2 :	Covering and packing problems
---------------	-------------------------------

G2c3 :	Knapsack problems
---------------	-------------------

G2c4 :	Matching problems
---------------	-------------------

G2c5 :	Routing, scheduling, location problems
---------------	--

G2c6 : Pure integer programming

NAG subprogram library

H02BAE Integer linear programming, Gomory's method with Wilson's cuts. Double precision version is H02BAF.

G2c7 : Mixed integer programming

G2d : Network (for network reliability search class M)

G2d1 : Shortest path

G2d2 : Minimum spanning tree

G2d3 : Maximum flow

G2d3a : Generalized networks

G2d3b : Networks with side constraints

G2d4 : Test problem generation

G2e : Quadratic programming

G2e1 : Positive definite Hessian (i.e. convex problem)

NAG subprogram library

H02AAE Quadratic programming, Beale's method. Double precision version is H02AAF.

G2e2 : Indefinite Hessian

G2f : Geometric programming

G2g : Dynamic programming

G2h : General nonlinear programming

G2h1 : Simple bounds

G2h1a : Smooth function

G2h1a1 : User provides no derivatives

IMSL subprogram library

ZXMWD Global minimum (with constraints) of a function of n variables.

NAG subprogram library

• **E04JAE** Minimum, function of several variables, simple bounds (easy-to-use), using function values only, quasi-Newton algorithm. Double precision version is **E04JAF**.

E04JBE Minimum, function of several variables, simple bounds (comprehensive), using function values only, quasi-Newton algorithm. Double precision version is **E04JBF**.

G2h1a2 : User provides first derivatives

NAG subprogram library

• **E04KAE** Minimum, function of several variables, simple bounds (easy-to-use), using first derivatives, quasi-Newton algorithm. Double precision version is **E04KAF**.

E04KBE Minimum, function of several variables, simple bounds (comprehensive), using first derivatives, quasi-Newton algorithm. Double precision version is **E04KBF**.

• **E04KCE** Minimum, function of several variables, simple bounds (easy-to-use), using first derivatives, modified Newton algorithm. Double precision version is **E04KCF**.

E04KDE Minimum, function of several variables, simple bounds (comprehensive), using first derivatives, modified Newton algorithm. Double precision version is **E04KDF**.

G2h1a3 : User provides first and second derivatives

NAG subprogram library

• **E04LAE** Minimum, function of several variables, simple bounds (easy-to-use), using first and second derivatives, modified Newton algorithm. Double precision version is **E04LAF**.

E04LBE Minimum, function of several variables, simple bounds (comprehensive), using first and second derivatives, modified Newton algorithm. Double precision version is **E04LBF**.

G2h1b : General function (no smoothness assumed)

G2h2 : Linear equality or inequality constraints

G2h2a : Smooth function

G2h2a1 : User provides no derivatives

G2h2a2 : User provides first derivatives

G2h2a3 : User provides first and second derivatives

G2h2b : General function (no smoothness assumed)

G2h3 : Nonlinear constraints

G2h3a : Equality constraints only

G2h3a1 : Smooth function and constraints

G2h3a1a : User provides no derivatives

G2h3a1b : User provides first derivatives of function and constraints

G2h3a1c : User provides first and second derivatives of function and constraints

G2h3a2 : General function and constraints (no smoothness assumed)

G2h3b : Equality and inequality constraints

G2h3b1 : Smooth function and constraints

G2h3b1a : User provides no derivatives

NAG subprogram library

E04UAE Minimum, function of n variables, non-linear constraints, function and constraint values only, sequential augmented Lagrangian quasi-Newton method. Double precision version is E04UAF.

G2h3b1b : User provides first derivatives of function and constraints

NAG subprogram library

E04VAE Minimum, function of several variables, general non-linear constraints, using first derivatives, sequential augmented Lagrangian quasi-Newton method. Double precision version is E04VAF.

E04VBE Minimum, function of several variables, general non-linear constraints, using first derivatives, sequential

augmented Lagrangian modified Newton method. Double precision version is E04VBF.

G2h3b1c : User provides first and second derivatives of function and constraints

NAG subprogram library

E04WAE Minimum, function of n variables, non-linear constraints, first and second derivatives, sequential augmented Lagrangian modified Newton method. Double precision version is E04WAF.

G2h3b2 : General function and constraints (no smoothness assumed)

G2i : Global solution to nonconvex problems

G3 : Optimal control

G4 : Service routines

G4a : Problem input (e.g., matrix generation)

G4b : Problem scaling

G4c : Check user-supplied derivatives

NAG subprogram library

E04HCE Check user's routine calculating first derivatives of function. Double precision version is E04HCF.

E04HDE Check user's routine calculating second derivatives of function. Double precision version is E04HDF.

F04YAE Check user's routine calculating Jacobian matrix of first derivatives. Double precision version is E04YAF.

E04YBE Check user's routine calculating second derivative term in Hessian matrix of sum of squares. Double precision version is E04YBF.

E04ZAE Check user's routines calculating first derivatives of function and constraints. Double precision version is E04ZAF.

E04ZBE Check user's routines calculating second derivatives of function and constraints. Double precision version is E04ZBF.

G4d : Find feasible point

IMSL subprogram library

ZSRCH Generate points in an n dimensional space.

NAG subprogram library

H01AFE Find feasible point or vertex which satisfies linear constraints. Double precision version is H01AFF.

G4e : Check for redundancy

G4f : Other

NAG subprogram library

E04HBE Finite-difference intervals for estimating first derivatives. Double precision version is E04HBF.

H: Differentiation and Integration

This chapter contains programs for performing the fundamental operations of calculus, the computation of derivatives and the evaluation of definite integrals.

January 1984

H :	Differentiation, integration
------------	------------------------------

H1: Numerical differentiation

This class contains programs which approximate derivatives of a function (given analytically or by data values) at one or a few points. It does not include programs which first produce a fit if the fitted function can be used independently. Programs which do this are classified in chapters E and K.

Computing derivatives numerically often leads to serious rounding errors. If the function to be differentiated is available as a subprogram rather than as data values, adaptive procedures can be used to locate the best mesh to use in some interval. If the function can be extended to the complex plane then the most effective method is to represent the derivative by the Cauchy integral formula and use the trapezoidal quadrature rule to evaluate it.

January 1984

References

1. Lyness, J. N. and C. B. Moler (1967). Numerical Differentiation of Analytic Functions, *SIAM Journal on Numerical Analysis*, Vol. 4, No. 2, pp. 202-210.

H1 :	Numerical differentiation
-------------	---------------------------

IMSL subprogram library

DRVTE Calculate first, second, or third derivative of a user-supplied function.

MATHWARE subprogram library (OLIVER sublibrary)

DIFFERENTI Calculates 1st, 2nd, or 3rd derivative of given function F at fixed point x to specified accuracy, program picks evaluation points.

NAG subprogram library

D04AAE Numerical differentiation of a function of one real variable, derivatives up to order 14. Double precision version is D04AAF.

PORT subprogram library

CSPDI Finds a numerical approximation to the first derivative at requested points in given input data by using spline interpolation. Double precision version is DCSPDI.

H2: Quadrature (numerical evaluation of definite integrals)

This category includes routines which evaluate the definite integral of a function which is either given explicitly or by means of a table of data. Programs can either be automatic or non-automatic.

Most routines for the integration of an explicit function require that the integrand be in the form of a Fortran function subprogram. In this case the exact function name must appear

- (1) in an EXTERNAL statement in the user's main program,
- (2) in the CALL statement which invokes the quadrature subroutine, and
- (3) as a Fortran FUNCTION itself.

The one exception to this is the very easy to use module Q1DA, which assumes that the function is called F.

Users must always be careful that their integrand function be defined wherever it might be evaluated. This depends on the specific routine but many programs always evaluate the integrand at the endpoints (in one dimension). If the problem is singular (e.g., $1/\sqrt{x}$ at 0.0) the user must arbitrarily set the integrand value at the singularity and prevent a divide by zero from occurring. Similarly, if the integrand has an apparent singularity (e.g., $\sin x/x$ at 0.0) the user must set the integrand value properly at this point (for this example $f(0) = 1$ is appropriate). A more correct solution is to use a program which does not perform endpoint evaluation. Sometimes the documentation omits this information. If internal singularities occur this same type of difficulty will arise. If the location of any singularities are known, it is highly recommended that the range of integration be split into pieces with the singularities only at the endpoints. This can be done automatically by some programs, for example, Q1DAX or QAGP by simply telling the integrator the locations of any problem points.

For "one shot" problems adaptive methods are very efficient in terms of human time. Adaption means that the integrand evaluation points are selected by the program and will vary from problem to problem. Automatic non-adaptive programs are also common. Some of these use a sequence of meshes, increasingly fine, and combine the results using an extrapolation algorithm. For the most part these techniques are not as efficient as the adaptive ones.

All automatic programs incur a penalty in terms of overhead, or surcharge. Non-automatic routines depend on a quadrature evaluation rule, a formula. Gauss quadrature rules are very efficient in terms of integrand evaluations but do not easily lend themselves to estimating errors. These formulas involve irrational points and weights which must be computed and tabulated in advance to be efficient. However they can save substantial computing time if the user is prepared to spend some time thinking about his problem and experimenting with different numbers of points. A number of routines which can generate and retrieve these formulas are listed in H2c. Programs which evaluate the formulas are listed in H2a1a2, H2a2a2, H2a3a2, H2b2a2 and H2b2b2.

Kronrod quadrature is a relatively new development. This involves the use of a pair of formulas, an n -point Gauss and a $2n + 1$ point Kronrod of which n points are the original Gauss points. This allows for an accurate error estimate at modest cost. Many new programs use these formulas. See [3] for details.

There are several programs for the integration of tabular, or gridded, data. The underlying idea is that some function is "fit" to the data and the function is then integrated. Some programs accept the user's data directly, such as CSPQU. Others require that the user input the form of the function, usually either a spline or a polynomial, such as PCHIA or E02AJE. In the second case most users will have to precede the call to the integrator by a call to a "fitter", either an interpolator from Chapter E or an approximator from Chapter K.

Programs also exist for multiple integration. For the integration over hyperrectangles, one can achieve 5-6 figure accuracy for smooth functions and 2-3 figure accuracy for non-smooth functions up to dimension 15. For integration over a bounded irregular region, one embeds this region inside a hyperrectangle and defines the integrand to be zero outside the region. In such cases a Monte Carlo type program, such as D01GBE, is recommended. One can also integrate over hyperspheres up to dimension 4 and over simplexes.

Two dimensional adaptive quadrature programs are just beginning to come into use. The overhead surcharge factor is even higher and there is much more scope for creativity by the problem originator. It has been common practice to use a pair of automatic one-dimensional programs to calculate iteratively a 2-D integral. In such cases it is suggested that the inner integral be evaluated more accurately to avoid instability. A COMMON block is also required to pass the second independent variable to the inner integrand which must be a function of only one argument.

January 1984

References

1. Davis, P. J. and P. Rabinowitz (1984). *Methods of Numerical Integration*, 2nd edition, Academic Press, New York.
2. Lyness, J. N. (1977). Quid, Quo, Quadrature?, in *The State of the Art in Numerical Analysis*, D. Jacobs (Editor), Academic Press, New York, pp. 535-560.
3. Piessens, R. et. al. (1983) *QUADPACK*, Springer-Verlag, Berlin.
4. Stroud, A. (1971). *Approximate Calculation of Multiple Integrals*, Prentice-Hall, Englewood Cliffs, New Jersey.

H2 : Quadrature (numerical evaluation of definite integrals)

H2a : One-dimensional integrals

H2a1 : Finite interval (general integrand)

H2a1a : Integrand available via user-defined procedure

H2a1a1 : Automatic (user need only specify required accuracy)

CMLIB subprogram library (Q1DA sublibrary)

- **Q1DA** Automatic evaluation of a user-defined function of one variable. Special features include randomization and singularity weakening.
- Q1DAX** Flexible subroutine for the automatic evaluation of definite integrals of a user-defined function of one variable. Special features include randomization, singularity weakening, restarting, specification of an initial mesh (optional), and output of smallest and largest integrand values.
- Q1DB** Automatic evaluation of a user-defined function of one variable. Integrand must be a Fortran Function but user may select name. Special features include randomization and singularity weakening. Intermediate in usage difficulty between Q1DA and Q1DAX.

CMLIB subprogram library (QUADSP sublibrary)

- **QAG** Automatic adaptive integrator, will handle many non-smooth integrands using Gauss Kronrod formulas. Double precision version is DQAG.
- QAGE** Automatic adaptive integrator, can handle most non-smooth functions also provides more information than QAG. Double precision version is DQAGE.
- **QAGS** Automatic adaptive integrator, will handle most non-smooth integrands including those with endpoint singularities, uses extrapolation. Double precision version is DQAGS.
- QAGSE** Automatic adaptive integrator, can handle integrands with endpoint singularities provides more information than QAGS. Double precision version is DQAGSE.
- QNG** Automatic non-adaptive integrator for smooth functions, using Gauss Kronrod Patterson formulas. Double precision version is DQNG.

IMSL subprogram library

- DCADRE** Numerical integration of a function using cautious adaptive Romberg extrapolation.

NAG subprogram library

- D01AHE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval suitable for well-behaved integrands. Double precision version is D01AHF.
- D01AJE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval allowing for badly-behaved integrands. Double precision version is D01AJF.
- D01ARE** Computes definite integral over a finite range to a specified relative or absolute accuracy, using Patterson's method. Double precision version is D01ARF.
- D01BDE** Quadrature for one-dimensional integrals, non-adaptive integration over a finite interval. Double precision version is D01BDF.

PORT subprogram library

- ODEQ** Finds the integral of a set of functions over the same interval by using the differential equation solver ODES1. For smooth functions. Double precision version is DODEQ.
- **QUAD** Finds the integral of a general user defined EXTERNAL function by an adaptive technique to given absolute accuracy. Double precision version is DQUAD.
- RQUAD** Finds the integral of a general user defined EXTERNAL function by an adaptive technique. Combined absolute and relative error control. Double precision version is DRQUAD.

H2a1a2 : Nonautomatic

CMLIB subprogram library (QUADSP sublibrary)

- QK15** Evaluates integral of given function on an interval with a 15 point Gauss Kronrod formula and returns error estimate. Double precision version is DQK15.
- QK21** Evaluates integral of given function on an interval with a 21 point Gauss Kronrod formula and returns error estimate. Double precision version is DQK21.
- QK31** Evaluates integral of given function on an interval with a 31 point Gauss Kronrod formula and returns error estimate. Double precision version is DQK31.
- QK41** Evaluates integral of given function on an interval with a 41 point Gauss Kronrod formula and returns error estimate. Double precision version is DQK41.
- QK51** Evaluates integral of given function on an interval with a 51 point Gauss Kronrod formula and returns error estimate. Double precision version is DQK51.
- QK61** Evaluates integral of given function on an interval with a 61 point Gauss Kronrod formula and returns error estimate. Double precision version is DQK61.

NAG subprogram library

- D01BAE** Quadrature for one-dimensional integrals, Gaussian rule-evaluation. Double precision version is D01BAF.

H2a1b : Integrand available only on grid

H2a1b1 : Automatic (user need only specify required accuracy)

H2a1b2 : Nonautomatic

NAG subprogram library

- D01GAE** Quadrature for one-dimensional integrals, integration of a function defined by data values only. Double precision version is D01GAF.

PORT subprogram library

- CSPQU** Finds the integral of a function defined by pairs (x,y) of input points. The x's can be unequally spaced. Uses spline interpolation. Double precision version is DCSPQU.

H2a2 : Finite interval (special integrand, e.g. weight functions, oscillating, singular, principal value, splines, etc.)

H2a2a : Integrand available via user-defined procedure

H2a2a1 : Automatic (user need only specify required accuracy)

CMLIB subprogram library (BSPLINE sublibrary)

- BFQAD** Integrates function times derivative of B-spline from X1 to X2. The B-spline is in "B" representation. Double precision version is DBFQAD.
- PFQAD** Computes integral on (X1,X2) of product of function and the ID-th derivative of B-spline which is in

piecewise polynomial representation. Double precision version is DPFQAD.

CMLIB subprogram library (QUADSP sublibrary)

- **QAGP** Automatic adaptive integrator, allows user to specify location of singularities or difficulties of integrand, uses extrapolation. Double precision version is DQAGP.
- QAGPE** Automatic adaptive integrator for function with user specified endpoint singularities, provides more information than QAGP. Double precision version is DQAGPE.
- **QAWC** Cauchy principal value integrator, using adaptive Clenshaw Curtis method (real Hilbert transform). Double precision version is DQAWC.
- QAWCE** Cauchy Principal value integrator, provides more information than QAWC (real Hilbert transform). Double precision version is DQAWCE.
- **QAWO** Automatic adaptive integrator for integrands with oscillatory sin or cosine factor. Double precision version is DQAWO.
- QAWOE** Automatic integrator for integrands with explicit oscillatory sin or cosine factor, provides more information than QAWO. Double precision version is DQAWOE.
- **QAWS** Automatic integrator for functions with explicit algebraic and/or logarithmic endpoint singularities. Double precision version is DQAWS.
- QAWSE** Automatic integrator for integrands with explicit algebraic and/or logarithmic endpoint singularities, more information than QAWS. Double precision version is DQAWSE.
- QMOMO** Computes integral of k-th degree Tchebycheff polynomial times selection of functions with various singularities. Double precision version is DQMOMO.

NAG subprogram library

- D01AKE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, method suitable for oscillating functions. Double precision version is D01AKF.
- D01ALE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, allowing for singularities at user-specified points. Double precision version is D01ALF.
- D01ANE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, weight function $\cos(\omega x)$ or $\sin(\omega x)$. Double precision version is D01ANF.
- D01APE** Adaptive integration of a function of one variable over a finite interval with weight function with algebraico-logarithmic endpoint singularities. Double precision version is D01APF.
- D01AQE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, weight function $1/(x-c)$ (Hilbert transform). Double precision version is D01AQF.

PORT subprogram library

- BQUAD** Adaptively integrates functions which have discontinuities in their derivatives. User can specify these points. Double precision version is DBQUAD.

H2a2a2 : Nonautomatic

CMLIB subprogram library (QUADSP sublibrary)

- QC25C** Uses 25 point Clenshaw-Curtis formula to estimate integral of $F \cdot W$ where $W=1/(X-C)$. Double precision version is DQC25C.
- QC25F** Clenshaw-Curtis integration rule for function with cos or sin factor, also uses Gauss Kronrod formula. Double precision version is DQC25F.
- QC25S** Estimates integral of function with algebraico-logarithmic singularities with 25 point Clenshaw-Curtis formula and gives error estimate. Double precision version is DQC25S.
- QK15W** Evaluates integral of given function times arbitrary weight function on interval with 15 point Gauss Kronrod formula and gives error estimate. Double precision version is DQK15W.

NAG subprogram library

D01FBE Quadrature for 1 to 20 dimensional integrals over a hyper-rectangle, Gaussian rule-evaluation, extensive selection. Double precision version is D01FBF.

H2a2b : Integrand available only on grid

H2a2b1 : Automatic (user need only specify required accuracy) or exact

CMLIB subprogram library (BSPLINE sublibrary)

BSQAD Computes the integral of a B-spline from X1 to X2. The B-spline must be in "B" representation. Double precision version is DBSQAD.

PPQAD Computes the integral of a B-spline from X1 to X2. The B-spline must be in piecewise polynomial representation. Double precision version is DPPQAD.

CMLIB subprogram library (PCHIP sublibrary)

PCHIA Evaluates the definite integral of a piecewise cubic Hermite function over an arbitrary interval.

PCHID Evaluates the definite integral of a piecewise cubic Hermite function over an interval whose endpoints are data points.

IMSL subprogram library

DCSQDU Cubic spline quadrature.

NAG subprogram library

E02AJE Integral of fitted polynomial in Chebyshev series form. Double precision version is E02AJF.

E02BDE Definite integral of a cubic previously computed by E02BAE in its "B" representation. Double precision version is E02BDF.

PORT subprogram library

BSPLI Obtains the integrals of basis splines, from the left-most mesh point to a specified set of points. Double precision version is DBSPLI.

SPLNI Integrates a function described previously by an expansion in terms of B-splines. Several integrations can be performed in one call. Double precision version is DSPLNI.

H2a2b2 : Nonautomatic

H2a3 : Semi-infinite interval (integrand can include e^{-x} weight function)

H2a3a : Integrand available via user-defined procedure

H2a3a1 : Automatic (user need only specify required accuracy)

CMLIB subprogram library (QUADSP sublibrary)

• **QAGI** Automatic adaptive integrator for semi-infinite or infinite intervals. Uses nonlinear transformation and extrapolation. Double precision version is DQAGI.

- QAGIE** Automatic integrator for semi-infinite or infinite intervals and general integrands, provides more information than QAGI. Double precision version is DQAGIE.
- **QAWF** Automatic integrator for Fourier integrals on (a, ∞) with factors $\sin(\omega x)$ or $\cos(\omega x)$, by integrating between zeros. Double precision version is DQAWF.
- QAWFE** Automatic integrator for Fourier integrals, with $\sin(\omega x)$ factor on (a, ∞) , provides more information than QAWF. Double precision version is DQAWFE.

NAG subprogram library

- D01AME** Quadrature for one-dimensional integrals, adaptive integration of a function over an infinite or semi-infinite interval. Double precision version is D01AMF.

H2a3a2 : Nonautomatic

CMLIB subprogram library (QUADSP sublibrary)

- QK15I** Evaluates integral of given function on semi-infinite or infinite interval with a transformed 15 point Gauss Kronrod formula and gives error estimate. Double precision version is DQK15I.

NAG subprogram library

- D01BAE** Quadrature for one-dimensional integrals, Gaussian rule-evaluation. Double precision version is D01BAF.
- D01FBE** Quadrature for 1 to 20 dimensional integrals over a hyper-rectangle, Gaussian rule-evaluation, extensive selection. Double precision version is D01FBF.

H2a4 : Infinite interval (integrand can include e^{-x^2} weight function)

H2a4a : Integrand available via user-defined procedure

H2a4a1 : Automatic (user need only specify required accuracy)

CMLIB subprogram library (QUADSP sublibrary)

- **QAGI** Automatic adaptive integrator for semi-infinite or infinite intervals. Uses nonlinear transformation and extrapolation. Double precision version is DQAGI.
- QAGIE** Automatic integrator for semi-infinite or infinite intervals and general integrands, provides more information than QAGI. Double precision version is DQAGIE.

NAG subprogram library

- D01AME** Quadrature for one-dimensional integrals, adaptive integration of a function over an infinite or semi-infinite interval. Double precision version is D01AMF.

H2a4a2 : Nonautomatic

CMLIB subprogram library (QUADSP sublibrary)

- QK15I** Evaluates integral of given function on semi-infinite or infinite interval with a transformed 15 point Gauss Kronrod formula and gives error estimate. Double precision version is DQK15I.

NAG subprogram library

- D01BAE** Quadrature for one-dimensional integrals, Gaussian rule-evaluation. Double precision version is D01BAF.
- D01FBE** Quadrature for 1 to 20 dimensional integrals over a hyper-rectangle, Gaussian rule-evaluation, extensive selection. Double precision version is D01FBF.

H2b : Multidimensional integrals

H2b1 : One or more hyper-rectangular regions

H2b1a : Integrand available via user-defined procedure

H2b1a1 : Automatic (user need only specify required accuracy)

IMSL subprogram library

- DMLIN** Numerical integration of a function of several variables over a hyper-rectangle (Gaussian method).

NAG subprogram library

- D01FAE** Quadrature for multi-dimensional integrals over a hyper-rectangle, Monte Carlo method. Double precision version is D01FAF.
- D01FCE** Quadrature for multi-dimensional integrals over a hyper-rectangle, adaptive method. Double precision version is D01FCF.
- D01GBE** Calculates an approximation to the integral of a function over a hyper-rectangular region, using a Monte-Carlo method. An approximate relative estimate is also returned. Suitable for low accuracy work. Double precision version is D01GBF.

H2b1a2 : Nonautomatic

NAG subprogram library

- D01FBE** Quadrature for 1 to 20 dimensional integrals over a hyper-rectangle, Gaussian rule-evaluation, extensive selection. Double precision version is D01FBF.
- D01FDE** Calculates an approximation to a definite integral in up to 30 dimensions, using the method of Sag and Szekeres. The region of integration is an n-sphere, or by built-in transformation via the unit n-cube, any product region. Double precision version is D01FDF.
- D01GCE** Calculates an approximation to a definite integral in up to 20 dimensions, using the Korobov-Conroy number theoretic method. Returns a simple error estimate by repeating the computation with different (randomized) sets of points. Double precision version is D01GCF.

H2b1b : Integrand available only on grid

H2b1b1 : Automatic (user need only specify required accuracy)

H2b1b2 : Nonautomatic

IMSL subprogram library

DBCQDU Bicubic spline quadrature.

H2b2 : Nonrectangular region, general region

H2b2a : Integrand available via user-defined procedure

H2b2a1 : Automatic (user need only specify required accuracy)

IMSL subprogram library

DBLIN Numerical integration of a function of two variables.

NAG subprogram library

D01DAE Quadrature for two-dimensional integrals over a finite region. Double precision version is **D01DAF**.

D01JAE Attempts to evaluate an integral over an n-dimensional sphere ($n=2,3,4$), to a user specified absolute or relative accuracy, by means of a modified Sag-Szekeres method. Can handle singularities on the surface or at the center of the sphere. Returns an error estimate. Double precision version is **D01JAF**.

H2b2a2 : Nonautomatic

NAG subprogram library

D01FDE Calculates an approximation to a definite integral in up to 30 dimensions, using the method of Sag and Szekeres. The region of integration is an n-sphere, or by built-in transformation via the unit n-cube, any product region. Double precision version is **D01FDF**.

D01PAE Returns a sequence of approximations to the integral of a function over a multi-dimensional simplex, together with an error estimate for the last approximation. Double precision version is **D01PAF**.

H2b2b : Integrand available only on grid

H2b2b1 : Automatic (user need only specify required accuracy)

H2b2b2 : Nonautomatic

H2c : Service routines (compute weight and nodes for quadrature formulas)

NAG subprogram library

D01BBE Weights and abscissae for Gaussian quadrature rules, restricted choice of rule, using pre-computed weights and abscissae. Double precision version is **D01BBF**.

D01BCE Weights and abscissae for Gaussian quadrature rules, more general choice of rule calculating the weights

and abscissae. Double precision version is D01BCF.

PORT subprogram library

- GAUSQ** Finds the abscissae and weights for Gauss quadrature on the interval (a,b) for a general weight function with known moments. Double precision version is DGAUSQ.
- GQ0IN** Finds the abscissae and weights for Gauss-Laguerre quadrature on the interval (0,+infinity). Double precision version is DGQ0IN.
- GQM11** Finds the abscissae and weights for Gauss-Legendre quadrature on the interval (-1,1). Double precision version is DGQM11.

I: Differential and Integral Equations

Differential and integral equations are the basis of most mathematical models of continuous processes, and hence the solution of these equations is a very important problem found in many applications. This chapter is divided into three parts which reflect the most general types of such equations—ordinary differential equations (ODEs), partial differential equations (PDEs), and integral equations.

January 1984

I: Differential and integral equations

11: Ordinary differential equations

Physical laws are often posed in the form of systems of ordinary differential equations. Most programs for solving ODEs operate only on systems of first order equations of the form

$$y' = f(t, y)$$

or

$$Ay' = f(t, y)$$

where y and f are "vectors" and A is a square matrix. Most higher order systems can be reduced to one of these forms by a change of variable.

An **initial value problem** is a system of ODEs and a vector of numbers which specify the solution y at a specific (initial) time. The problem is to determine the solution at subsequent times. An initial value problem is **stiff** if the physical system contains time constants varying over several decades and the solution is desired in a range where the fastest components have died out. A **boundary value problem** is similar except that the "data" is not all given at a single point, e.g., some components may be specified at t_1 and others at t_2 etc.

Programs for solving initial or boundary value problems usually approximate the solution at a discrete set of points which are chosen dynamically by the integrator. These points may not correspond to the user's output points, rather they are selected to get to the end as efficiently as possible. This requires an interpolation procedure (invisible to the user). The solution methods are often implicit and hence ultimately require the solution of systems of linear algebraic equations. Boundary value problems are more difficult and the programs are less reliable. Current techniques include shooting (solving a sequence of initial value problems), collocation (forcing an approximate solution to satisfy the ODEs at selected points), integral equations, invariant imbedding and finite differences.

Most programs require an error tolerance to be specified by the user. If the program terminates normally the implication is that the solution is given to within that tolerance. In practice this is often true. However, the input tolerance is usually used only to control local errors. There is no attempt to control accumulation of error over a long sequence of integration steps.

In Fortran subprograms that allow the user to specify the name of a subroutine to evaluate the differential equations be sure that this name appears

- (1) in an EXTERNAL statement in the main program,
- (2) in the CALL statement which invokes the ODE solver, and
- (3) as the name of a SUBROUTINE.

Another frequent programming error is to confuse the array Y which is input to the main ODE program with the array of the same name in the user-supplied subroutine to evaluate the derivatives.

January 1984

References

1. Childs, B., M. Scott, J. W. Daniel, et al. (Editors) (1979). *Codes for Boundary-Value Problems in Ordinary Differential Equations*, Springer-Verlag, New York.
2. Gladwell, I. (1979). Initial Value Routines in the NAG Library, *ACM Transactions on Mathematical Software*, Vol. 5, No. 4, pp. 386-400.
3. Keller, H. B. (1976). *Numerical Solution of Two Point Boundary Value Problems*, SIAM, Philadelphia.
4. Lambert, J. D. (1973). *Computational Methods in Ordinary Differential Equations*, John Wiley & Sons, New York.
5. Prenter, P. M. (1975). *Splines and Variational Methods*, John Wiley & Sons, New York.

6. Shampine, L. F. and M. K. Gordon (1975). *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*, W. H. Freeman and Co., San Francisco.

7. Shampine, L. F., H. A. Watts, and S. M. Davenport (1976). Solving Nonstiff Ordinary Differential Equations - The State of the Art, *SIAM Review*, Vol. 18, No. 3, pp. 376-411.

8. Shampine, L. F. and C. W. Gear (1979). A User's View of Solving Stiff Ordinary Differential Equations, *SIAM Review*, Vol. 21, No. 1, pp. 1-17.

II :	Ordinary differential equations
-------------	---------------------------------

IIa :	Initial value problems
--------------	------------------------

PLOD interactive system

- **PLOD** An easy to use interactive system for the solution of initial value problems for ordinary differential equations. Requires a Tektronix or Hewlett-Packard graphics terminal. The user can change initial conditions, interval, parameters etc., and examine various plots on the terminal. Little programming needed.

IIa1 :	General, nonstiff or mildly stiff
---------------	-----------------------------------

IIa1a :	One-step methods (e.g., Runge-Kutta)
----------------	--------------------------------------

CMLIB subprogram library (DEPAC sublibrary)

- DERKF** Solves a system of first order ordinary differential equations with arbitrary initial conditions by a Runge-Kutta method.

IMSL subprogram library

- DVERK** Differential equation solver - Runge Kutta-Verner fifth and sixth order method.

NAG subprogram library

- **D02BAE** Initial value problems for system of ordinary differential equations, (simple driver) Runge-Kutta-Merson method, over a range. Double precision version is D02BAF.
- **D02BBE** Initial value problems for system of ordinary differential equations, (simple driver) Runge-Kutta-Merson method, over a range with intermediate output. Double precision version is D02BBF.
- **D02BDE** Initial value problems for system of O.D.E.s, (simple driver) Runge-Kutta-Merson method, over a range with global error estimate and stiffness check. Double precision version is D02BDF.
- **D02BGE** Initial value problems for system of O.D.E.s, (simple driver) Runge-Kutta-Merson method, until a component of the solution attains a given value. Double precision version is D02BGF.
- **D02BHE** Initial value problems for system of O.D.E.s, (simple driver) Runge-Kutta-Merson method, until a function of the solution is zero. Double precision version is D02BHF.
- D02PAE** Initial value problems for system of O.D.E.s, integrating over a range (facilities for error-control and interrupts) Runge-Kutta-Merson method. Double precision version is D02PAF.
- D02YAE** Initial value problems for system of ordinary differential equations, integration over one step by Runge-Kutta-Merson method. Double precision version is D02YAF.

IIa1b : Multistep methods (e.g., Adams' predictor-corrector)**CMLIB subprogram library (CDRIV sublibrary)**

- **CDRIV1** Numerical integration of complex initial value problems for ordinary differential equations, Gear stiff formulas, Easy to use.
- CDRIV2** Numerical integration of complex initial value problems for ordinary differential equations, Gear stiff and Adams formulas, root finding.
- CDRIV3** Numerical integration of complex initial value problems for ODEs, Gear and Adams formulas, Implicit eqs., Sparse Jacobians, root finding.

CMLIB subprogram library (DEPAC sublibrary)

- DEABM** Solves a system of first order ordinary differential equations with arbitrary initial conditions by a predictor-corrector method.

CMLIB subprogram library (SDASSL sublibrary)

- SDASSL** Solves the system of differential/algebraic equations of the form $g(t,y, y_{\text{prime}})=0$, with given initial values. Double precision version is DDASSL.

CMLIB subprogram library (SDRIV sublibrary)

- **SDRIV1** Numerical Integration, Initial Value Problems, Ordinary Differential Eqs., Gear Stiff Formulas, Easy to Use. Double precision version is DDRIV1.
- SDRIV2** Numerical Integration, Initial Value Problems, Ordinary Differential Eqs., Gear / Adams Formulas. Double precision version is DDRIV2.
- SDRIV3** Numerical Integration, Initial Value Problems, Ordinary Differential Eqs., Implicit Eqs., Sparse Jacobians. Double precision version is DDRIV3.

IMSL subprogram library

- DGEAR** Differential equation solver - variable order Adams predictor corrector method or Gear's method.

NAG subprogram library

- **D02CAE** Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Adams method, over a range. Double precision version is D02CAF.
- **D02CBE** Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Adams method, over a range with intermediate output. Double precision version is D02CBF.
- **D02CGE** Initial value problems for system of O.D.E.s, (simple driver) variable order and step Adams method, until a component of the solution attains a given value. Double precision version is D02CGF.
- **D02CHE** Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Adams method, until a function of the solution is zero. Double precision version is D02CHF.
- D02QAE** Initial value problems for system of O.D.E.s, integrating over a range (error- control and interrupts) variable-order -step Adams method. Double precision version is D02QAF.

SLDGL subprogram library

- SLGA1** Solves by difference methods an initial value problem for a system of ordinary differential equations. The stepsize and order of the method are chosen automatically to maintain prescribed local discretization error bounds.
- SLGA2** Solves by difference methods a mixed implicit/algebraic initial value problem for a system of ordinary differential equations. Stepsize and order are chosen automatically to maintain estimated local discretization error within prescribed bounds.
- SLGA3** Solves by difference methods an initial value problem for an explicit system of ordinary differential equations. Chooses stepsize and order of method to maintain estimate of local discretization error within prescribed bounds. Also provides values of first derivative at output points.

- SLGA4** Solves by difference methods a mixed algebraic/implicit initial value problem for a first order system of ordinary differential equations. chooses stepsize and order to maintain estimate of local discretization error within prescribed bounds. Also provides derivative at output points.

11a1c : Extrapolation methods (e.g., Bulirsch-Stoer)

IMSL subprogram library

- DREBS** Differential equation solver extrapolation method.

PORT subprogram library

- **ODES** Solves an initial value problem for a system of ordinary differential equations. Easy to use. Double precision version is DODES.
- ODES1** Solves an initial value problem for a system of ordinary differential equations. Allows great flexibility and user control. Double precision version is DODES1.

11a2 : Stiff and mixed algebraic- differential equations

CMLIB subprogram library (CDRIV sublibrary)

- **CDRIV1** Numerical integration of complex initial value problems for ordinary differential equations, Gear stiff formulas, Easy to use.
- CDRIV2** Numerical integration of complex initial value problems for ordinary differential equations, Gear stiff and Adams formulas, root finding.
- CDRIV3** Numerical integration of complex initial value problems for ODEs, Gear and Adams formulas, Implicit eqs., Sparse Jacobians, root finding.

CMLIB subprogram library (DEPAC sublibrary)

- DEBDF** Solves a system of first order stiff ordinary differential equations with arbitrary initial conditions by Gear's method.

CMLIB subprogram library (SDRIV sublibrary)

- **SDRIV1** Numerical Integration, Initial Value Problems, Ordinary Differential Eqs., Gear Stiff Formulas, Easy to Use. Double precision version is DDRIV1.
- SDRIV2** Numerical Integration, Initial Value Problems, Ordinary Differential Eqs., Gear / Adams Formulas. Double precision version is DDRIV2.
- SDRIV3** Numerical Integration, Initial Value Problems, Ordinary Differential Eqs., Implicit Eqs., Sparse Jacobians. Double precision version is DDRIV3.

IMSL subprogram library

- DGEAR** Differential equation solver - variable order Adams predictor corrector method or Gear's method.

NAG subprogram library

- **D02EAE** Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Gear method for stiff systems, over a range. Double precision version is D02EAF.
- **D02EBE** Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for stiff systems, over a range with intermediate output. Double precision version is D02EBF.
- **D02EGE** Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for stiff systems, until the solution attains a given value. Double precision version is D02EGF.
- **D02EHE** Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for

stiff systems, until a function of the solution is zero. Double precision version is D02EHF.

D02QBE Initial value problems for system of O.D.E.s, integrating over a range (error- control and interrupts) variable-order -step Gear method for stiff systems. Double precision version is D02QBF.

I1b : Multipoint boundary value problems

I1b1 : Linear

CMLIB subprogram library (BVSUP sublibrary)

BVSUP Solves a system of linear two-point boundary value problems using superposition, orthogonalization, and variable step integration.

NAG subprogram library

D02GBE Boundary-value problems for system of O.D.E.s, finite difference technique with deferred correction (Pereyra), general linear problem. Double precision version is D02GBF.

D02JAE Solves a regular linear two point boundary value problem for a single n-th order ordinary differential equation by a Chebyshev series using collocation and least squares. Double precision version is D02JAF.

D02JBE Boundary-value problems for system of O.D.E.s, collocation and least-squares, system of 1st order linear equations. Double precision version is D02JBF.

D02TGE Boundary-value problems for system of ordinary differential equations, collocation and least-squares, system of n-th order linear equations. Double precision version is D02TGF.

I1b2 : Nonlinear

IMSL subprogram library

DTPTB Solve a system of ordinary differential equations with boundary conditions at two points, using a multiple shooting method.

DVCPR Solve a system of ordinary differential equations with boundary conditions at two points, using a variable order, variable step size finite difference method with deferred corrections.

NAG subprogram library

D02AGE Solves two point boundary value problem for a system of ODEs using initial value techniques. Double precision version is D02AGF.

• **D02GAE** Boundary-value problems for system of O.D.E.s, finite difference technique with deferred correction (Pereyra), simple non-linear problem. Double precision version is D02GAF.

D02HAE Boundary-value problems for system of O.D.E.s, shooting and matching technique, boundary values to be determined. Double precision version is D02HAF.

D02HBE Boundary-value problems for system of O.D.E.s, shooting and matching technique, general parameters to be determined. Double precision version is D02HBF.

D02RAE Boundary-value problems for system of O.D.E.s, finite difference technique with deferred correction, general non-linear problem, continuation facility. Double precision version is D02RAF.

D02SAE Boundary-value problems for system of O.D.E.s, shooting and matching technique, general parameters to be determined, subject to extra algebraic equations. Double precision version is D02SAF.

SLDGL subprogram library

SLGR0 Solves by difference methods on a non-equidistant grid a second-order implicit system of ordinary differential equations with implicit boundary conditions prescribed. An estimate of the discretization

error is provided.

- SLGR1** Solves by difference methods on an equidistant grid an implicit system of second-order ordinary differential equations with prescribed implicit boundary conditions. An estimate of the discretization error is provided.
- SLGR2** Solves by difference methods on a non-equidistant grid an implicit system of second-order ordinary differential equations with implicit boundary values. For a given relative accuracy gridpoints and order of method are chosen automatically to minimize number of gridpoints.
- SLGR3** Solves by difference methods on an equidistant grid an implicit system of second-order ordinary differential equations with implicit boundary values. For a prescribed relative accuracy the order of the method is chosen to minimize the number of gridpoints.
- SLGR4** Solves by difference methods on a non-equidistant grid an implicit system of second-order ordinary differential equations with implicit boundary conditions. For a prescribed relative accuracy the gridpoints and order of the method are chosen to minimize the number of gridpoints.
- SLGR5** Solves by difference methods on a non-equidistant grid an implicit system of second-order ordinary differential equations with implicit boundary conditions. For a given relative accuracy the gridpoints and order of method are chosen to minimize number of gridpts. Suited for boundary layer problems.

11b3 : Eigenvalue (e.g., Sturm-Liouville)

NAG subprogram library

- D02AGE** Solves two point boundary value problem for a system of ODEs using initial value techniques. Double precision version is D02AGF.
- D02KAE** Second-order Sturm-Liouville problems, regular system, finite range, eigenvalue only. Double precision version is D02KAF.
- D02KDE** Second-order Sturm-Liouville problems, regular/singular system, finite/infinite range, eigenvalue only. Double precision version is D02KDF.
- D02KEE** Second-order Sturm-Liouville problems, regular/singular system, finite/infinite range, eigenvalue and eigenfunction. Double precision version is D02KEF.

11c : Service routines (e.g., interpolation of solutions, error handling)

NAG subprogram library

- D02XAE** Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02PAE, all components. Double precision version is D02XAF.
- D02XBE** Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02PAE, one component. Double precision version is D02XBF.
- D02XGE** Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02QAE or D02QBE, all components. Double precision version is D02XGF.
- D02XHE** Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02QAE or D02QBE, one component. Double precision version is D02XHF.

PORT subprogram library

- ODESE** Standard error subprogram for the routine ODES1. Double precision version is DODESE.
- ODESH** Default HANDLE routine for ODES. Used to access the results at the end of each integration time step. Double precision version is DODESH.

12: Partial differential equations

Partial differential equations are an important tool for researchers modelling continuous processes in all areas of science. Such complex models are rarely solvable using analytic techniques; instead, they provide some of the most challenging problems in all of scientific computing. Successful numerical methods must forge an effective synthesis of techniques from such diverse areas as approximation theory, numerical quadrature, and the numerical solution of linear and nonlinear algebraic equations.

The most common problems are "second order" and may be classified as either elliptic, parabolic or hyperbolic, although the most complex systems are combinations of these. A solution is required on some one, two or three dimensional domain, which may be bounded (with irregular boundaries adding geometrical headaches) or not.

Elliptic equations model steady-state phenomena, with the solution determined by conditions specified on the boundaries of the domain (Laplace's equation, $u_{xx} + u_{yy} = 0$, is the prototype). **Parabolic** problems add the element of time, with the solution at future times dependent upon the given solution at some initial time (the heat equation, $u_t = u_{xx}$, is the prototype). **Hyperbolic** problems are also of the initial-boundary value problem type, but are characterized by the finite propagation speed of data (the wave equation, $u_{tt} = u_{xx}$, is the prototype).

Because of the great diversity of problem characteristics, most programs for solving partial differential equations have been aimed at a specific problem in a specific applications area, with little possibility of easy extension to other problems. Thus, the state of general purpose software for partial differential equations is still in its infancy. The exceptions are in the areas of separable elliptic equations on rectangular domains and parabolic system solvers in one space dimension.

Most software begins with some finite dimensional approximation of the spatial part of the differential equation and boundary conditions. Two basic techniques are used—finite differences and finite elements. In finite differences, derivatives are directly approximated by difference quotients, leading to a system of algebraic equations whose solution yields values of the unknown quantities at a finite set of points. In finite elements, the solution is represented as a finite sum of known functions, each of which is zero on most of the domain. Variational techniques are then used to obtain a system of algebraic equations which determine the unknown coefficients of this finite sum. Finite elements have been tremendously successful in such application areas as structural engineering where their ability to easily conform to complex geometries is essential. In other areas such as fluid dynamics, finite differences remain popular due to their inherent simplicity. The computational complexity of the problem increases nonlinearly with dimension, and storage and computation times may be prohibitive for all but the coarsest of approximations in three dimensions.

The most widely distributed package of subroutines for elliptic problems is undoubtedly the FISHPAK collection (available in CMLIB). These routines provide efficient and reliable solutions to separable elliptic problems. These are single linear elliptic equations which are defined as the sum of two one-dimensional equations, one depending only upon x and one depending only upon y . The domain must be rectangular (in Cartesian, polar, or surface spherical coordinates) with simple boundary conditions. In this case very fast techniques related to the numerical separation of variables may be applied. The SLDGL library contains several subprograms which discretize and solve general nonlinear systems of elliptic equations defined on rectangular domains. A single routine for problems on non-rectangular domains is also available. Each of these programs is based on finite differences. Laplace's equation is easily reformulated so that integral equation techniques may be applied. This is especially attractive for problems on complicated domains. The NAG library has a subprogram based on such techniques.

Most successful general purpose software packages for parabolic equations use the so-called method of lines. Here, the spatial approximation is used to generate a system of ordinary differential equations which is solved by existing general purpose software for that problem. Each of the IMSL, NAG, PDELIB, and SLDGL libraries have subprograms based on this technique.

January 1984

References

1. Ames, W. F. (1977). *Numerical Methods for Partial Differential Equations*, 2nd Edition, Academic Press, New York.
2. Boisvert, R. F. and R. A. Sweet (1983). Mathematical Software for Elliptic Boundary Value Problems, in *Sources and Development of Mathematical Software* W. Cowell (Editor) Prentice-Hall, Englewood Cliffs, NJ.
3. Carrier, G. F. and C. E. Pearson (1976). *Partial Differential Equations*, Academic Press, New York.
4. Gladwell, I. and R. Wait (Editors) (1979). *A Survey of Numerical Methods for Partial Differential Equations*, Oxford University Press, London.
5. Machura, M. and R. A. Sweet (1980). A Survey of Software for Partial Differential Equations, *ACM Transactions on Mathematical Software*, Vol. 6, No. 4, pp. 461-488.
6. Mitchell, A. R. and R. Wait (1977). *The Finite Element Method in Partial Differential Equations*, John Wiley & Sons, New York.
7. Mitchell A. R. and D. F. Griffiths (1980). *The Finite Difference Method in Partial Differential Equations*, John

Wiley & Sons, New York.

8. Roache, P. J. (1976). *Computational Fluid Dynamics*, Revised Printing, Hermosa Publishers, Albuquerque, New Mexico.

I2 : Partial differential equations

I2a : Initial boundary value problems

I2a1 : Parabolic

I2a1a : One spatial dimension

IMSL subprogram library

DPDES Solve a system of partial differential equations of the form $UT = FCN(X,T, U,UX, UXX)$ using the method of lines with cubic Hermite polynomials.

NAG subprogram library

D03PAE P.D.E.s, parabolic, one space variable, method of lines, single equation. Double precision version is D03PAF.

D03PBE P.D.E.s, parabolic, one space variable, method of lines, simple system. Double precision version is D03PBF.

D03PGE P.D.E.s, parabolic, one space variable, method of lines, general system. Double precision version is D03PGF.

PDELIB subprogram library (PDELIB sublibrary)

MOL1D Solves systems of linear or nonlinear initial-boundary-value problems in one space dimension. Can solve hyperbolic equations with or without discontinuities, parabolic equations (including reaction-diffusion equations). Uses the method of lines based on equi-spaced finite differences. Graphical output available.

PDECOL Solves general nonlinear systems of initial-boundary-value problems in one space dimension with general boundary conditions. Spatial derivatives may be of at most second order. Uses method of lines based on collocation of B-spline basis functions.

SLDGL subprogram library

SL1P1 Solves a fully implicit difference scheme for a one-dimensional system of parabolic differential equations with general boundary conditions on a specified equidistant spatial grid using a specified order spatial method. Order and stepsize in time and error estimate are computed.

SL1P2 Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with initial conditions determined from the solution of an ode boundary value problem. User specifies equidistant grid and order in space, stepsize and order in time and error estimate computed.

SL1P3 Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with specified initial and boundary conditions. For a given relative accuracy an equidistant spatial grid and optimal order are computed as well as stepsize and order in time.

SL1P4 Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with initial conditions determined from the solution of a ode boundary value problem. Given a relative accuracy an equidistant spatial grid and optimal order are determined.

SL1P5 Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with specified initial and boundary conditions and non-equidistant spatial grid and spatial order. Stepsize and order in time are computed as well as an estimate of global error.

- SL1P6** Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with initial conditions determined from an ode boundary value problem. A non-equidistant spatial grid and spatial order are specified and the stepsize and order in time and error estimate are computed.
- SL1P7** Solves a fully implicit difference scheme for a system of one-dimensional parabolic equations with specified initial and boundary conditions. For a specified relative accuracy a non-equidistant spatial grid and optimal order are computed as well as an estimate of global error.
- SL1P8** Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with initial conditions determined from an ode boundary value problem. For a given relative accuracy a non-equidistant spatial grid and order and error estimate are computed.

I2a1b : Two or more spatial dimensions

PDELIB subprogram library (PDELIB sublibrary)

- PDE TWO** Solves general nonlinear systems of initial-boundary-value problems in two spatial dimensions with quasi-linear boundary conditions. Uses the method of lines based upon finite differences on a user-specified rectangular mesh.

SLDGL subprogram library

- SL2P1** Solves a fully implicit difference scheme for an implicit system of two-dimensional parabolic equations on a rectangle with specified initial and boundary conditions. User provides non-equidistant spatial grid and spatial orders and stepsize and order in time and error estimate are computed.
- SL2P2** Solves a fully implicit difference scheme for an implicit system of two-dimensional parabolic equations. Initial conditions are computed from an elliptic boundary value problem. User specifies non-equidistant spatial grid and spatial orders and stepsize and order in time and error estimate are computed.
- SL2P3** Solves a fully implicit difference scheme for an implicit system of parabolic equations with given initial and boundary conditions. For a given error tolerance an optimal combination of spatial gridpoint distribution and order are computed as well as an estimate of the global error.
- SL2P4** Solves a fully implicit difference scheme for an implicit system of two-dimensional parabolic equations with the initial conditions determined as the solution of an elliptic boundary value problem. For a given tolerance optimal spatial gridpoint distribution and order are computed.
- SL3P1** Solves a fully implicit difference scheme for a three-dimensional system of parabolic equations with specified initial and boundary conditions on a parallelepiped. The user provides a non-equidistant spatial grid and spatial orders and stepsize and order in time and an error estimate are computed.

I2a2 : Hyperbolic

IMSL subprogram library

- DPDES** Solve a system of partial differential equations of the form $UT = FCN(X,T, U,UX, UXX)$ using the method of lines with cubic Hermite polynomials.

PDELIB subprogram library (PDELIB sublibrary)

- MOL1D** Solves systems of linear or nonlinear initial-boundary-value problems in one space dimension. Can solve hyperbolic equations with or without discontinuities, parabolic equations (including reaction-diffusion equations). Uses the method of lines based on equi-spaced finite differences. Graphical output available.

I2b : Elliptic boundary value problems

I2b1 : Linear

I2b1a : Second order

I2b1a1 : Poisson (Laplace) or Helmholtz equation

I2b1a1a : Rectangular domain (or topologically rectangular in the coordinate system)

CMLIB subprogram library (FSHPK sublibrary)

- HSTCRT** Solves the Helmholtz or Poisson equations in two dimensions in Cartesian coordinates on a staggered grid.
- HSTCSP** Solves a modified Helmholtz equation in spherical coordinates with axisymmetry using a staggered grid.
- HSTCYL** Solves a modified Helmholtz equation in cylindrical coordinates on a staggered grid.
- HSTPLR** Solves the Helmholtz or Poisson equation in polar coordinates on a staggered grid.
- HSTSSP** Solves the Helmholtz or Poisson equation in spherical coordinates on the surface of a sphere using a staggered grid.
- HW3CRT** Solves the Helmholtz or Poisson equation in three dimensions using Cartesian coordinates.
- HWSCRT** Solves the Helmholtz or Poisson equation in two dimensions in Cartesian coordinates.
- HWSCSP** Solves a modified Helmholtz equation in spherical coordinates with axisymmetry.
- HWSCYL** Solves a modified Helmholtz equation in cylindrical coordinates.
- HSWPLR** Solves the Helmholtz or Poisson equation in polar coordinates.
- HWSSSP** Solves the Helmholtz or Poisson equation in spherical coordinates on the surface of a sphere.

I2b1a1b : Nonrectangular domain

NAG subprogram library

- D03EAE** Partial differential equations, elliptic, Laplace's equation in 2-d for an arbitrary domain. Double precision version is D03EAF.

I2b1a2 : Other separable problems

CMLIB subprogram library (FSHPK sublibrary)

- SEPCLI** Solves separable elliptic boundary value problems on a rectangle.
- SEPXL** Solves separable elliptic boundary value problems on a rectangle with constant coefficients in one direction.

I2b1a3 : Nonseparable problems

I2b1c : Higher order equations (e.g., biharmonic)

I2b2 : Nonlinear**SLDGL subprogram library**

- SL2E2** Solves by a fully implicit difference scheme a two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. For a prescribed relative accuracy the gridpoints and order of the method are automatically determined.
- SL2E4** Solves by a fully implicit difference scheme a two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. A non-equidistant grid and the order of the method are prescribed. An estimate of the discretization error is computed.
- SL2E5** Solves by a fully implicit difference scheme a two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. For a prescribed relative accuracy the non-equidistant grid and order of the method are determined automatically. Discretization error estimated.
- SL2E6** Solves by a fully implicit difference scheme a two-dimensional elliptic boundary value problem with general boundary conditions on a rectangle. For a prescribed equidistant grid and relative accuracy the routine automatically determines the order of the method. Error estimate provided.
- SL2E7** Solves a fully implicit difference scheme for two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. A non-equidistant grid and the order of the method are prescribed. An estimate of the discretization error is provided. Line iteration used.
- SL2E8** Solves a fully implicit difference scheme for a two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. For a prescribed relative accuracy the gridpoints and order of the method are determined. Approximation solved by line iteration.
- SL2EB1** Solves a fully implicit difference scheme for a system of two-dimensional elliptic equations on a general region with Dirichlet or Neumann boundary conditions specified. A non-equidistant grid and the order of the method are prescribed. An estimate of the discretization error is provided.
- SL3E1** Solves a fully implicit difference scheme for a system of three-dimensional elliptic equations on a parallelepiped with general boundary conditions. A non-equidistant grid and the order of the method are prescribed. An estimate of the discretization error is computed.

I2b3 : Eigenvalue**I2b4 :** Service routines**I2b4a :** Domain triangulation (*search also class P2a2c1*)**NAG subprogram library**

- D03MAE** Triangulation of a plane region. Double precision version is D03MAF.

I2b4b : Solution of discretized elliptic equations**CMLIB subprogram library (FSHPK sublibrary)**

- BLKTRI** Solves block tridiagonal systems of linear algebraic equations arising from the discretization of separable elliptic partial differential equations.
- CBKTR** Solves certain complex block tridiag. systems of lin. eqns. arising from the discretization of separable elliptic partial differential equations.
- CMGNBN** Solves certain complex block tridiag. systems of lin. eqns. arising from Helmholtz or Poisson eqn in 2 dim. Cartesian coordinates.

- GENBUN** Solves certain block tridiagonal systems of lin. equations arising from Helmholtz or Poisson equation in two Cartesian coordinates.
- POIS3D** Solves block tridiagonal linear systems of algebraic equations arising from the discretization of separable elliptic partial differential equations in 3D.
- POISTG** Solves block tridiagonal linear systems of algebraic equations arising from the discretization of separable elliptic partial differential equations.

MATHWARE subprogram library (ITPACK sublibrary)

- JCG** Iterative solution of large sparse systems of linear equations. Jacobi method, conjugate gradient acceleration, adaptive parameter selection.
- JSI** Iterative solution of large sparse systems of linear equations. Jacobi method, Chebyshev acceleration, adaptive parameter selection.
- RSCG** Iterative solution of large sparse systems of linear equations. Reduced system method, conjugate gradient acceleration, adaptive.
- RSSI** Iterative solution of large sparse systems of linear equations. Reduced system method, Chebyshev acceleration, adaptive.
- SOR** Iterative solution of large sparse systems of linear equations. SOR method, adaptive parameter selection.
- SSORCG** Iterative solution of large sparse systems of linear equations. SSOR method, conjugate gradient acceleration, adaptive parameter selection.
- SSORSI** Iterative solution of large sparse systems of linear equations. SSOR method, Chebyshev acceleration, adaptive parameter selection.

NAG subprogram library

- D03EBE** P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 5-point 2-d molecule, iterate to convergence. Double precision version is D03EBF.
- D03ECE** P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 7-point 3-d molecule, iterate to convergence. Double precision version is D03ECF.
- D03UAE** P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 5-point 2-d molecule, one iteration. Double precision version is D03UAF.
- D03UBE** P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 7-point 3-d molecule, one iteration. Double precision version is D03UBF.

13: Integral equations

Integral equations may be "first" or "second" kind depending on whether the unknown function appears only inside or both inside and outside the integral. The latter are much easier to solve, both computationally and mathematically. First kind equations are often very sensitive numerically and special care must be exercised or the results will be meaningless. Easier problems correspond to those with more singular kernels; very smooth kernels are particularly difficult to handle.

January 1984

References

1. Baker, C. T. H. (1977). *The Numerical Treatment of Integral Equations*, Oxford University Press, New York.
2. Muskhelishvili, N. I. (1958). *Singular Integral Equations: Boundary Problems of Functions Theory and Their Applications to Mathematical Physics*, Revised Edition, Wolters-Noordhoff Publishing, Groningen, The Netherlands.
3. Noble, B. (1977). The Numerical Solution of Integral Equations, in *The State of the Art in Numerical Analysis*, D. Jacobs (Editor), Academic Press, New York, pp. 915-986.

I3 : Integral equations**NAG subprogram library**

- D05AAE** Linear non-singular Fredholm integral equation, 2nd kind, split kernel. Double precision version is D05AAF.
- D05ABE** Linear non-singular Fredholm integral equation, 2nd kind, smooth kernel. Double precision version is D05ABF.

J: Integral Transforms

Programs in this chapter compute integral transforms, the most common of which is the Fourier transform. FFT programs allow transformation to/from spectral space in an amount of computer time proportional to $n \ln(n)$ where n is the number of data points. The programs are most efficient when n is highly composite, a power of two being the most propitious choice. It may take 200-300% more computing for an $n = 127$ transform than an $n = 128$, although most programs will work correctly for any n . On the other hand, adding zero value data points can introduce spurious effects in the spectrum. The program selected must not only satisfy the physical model requirements (pure cosine transform, complex transform, etc.) but should also be appropriate for the intended computer. FFT programs can be made extra efficient by taking advantage of special machine hardware (e.g. vectorization capability) or by writing segments in assembly language. Even in Fortran, implementations on the same machine can easily differ 30% in running time. Note that while there are a few multidimensional FFT programs, most one dimensional modules are easily used in a multidimensional setting by writing a simple driver.

January 1984

References

1. Brigham, E. O. (1974). *The Fast Fourier Transform*, Prentice-Hall, Englewood Cliffs, New Jersey.
2. Digital Signal Processing Committee (1979). *Programs for Digital Signal Processing*, IEEE Press, New York.

J :	Integral transforms
------------	---------------------

J1 :	Fast Fourier transforms (<i>search class L10 for time series analysis</i>)
-------------	--

J1a :	One-dimensional
--------------	-----------------

J1a1 :	Real
---------------	------

CMLIB subprogram library (FFTPKG sublibrary)

- **EZFFTB** Backward real discrete (fast) Fourier transform. Performs Fourier synthesis. Easy to use version.
- **EZFFTF** Forward real discrete (fast) Fourier transform. Performs Fourier analysis. Easy to use version.
- RFFTB** Computes real periodic sequence from real Fourier coefficients. Performs Fourier synthesis.
- RFFTF** Computes Fourier coefficients of real periodic sequence (fast). Performs Fourier analysis.

IMSL subprogram library

- FFTRC** Compute the fast Fourier transform of a real valued sequence.

NAG subprogram library

- C06EAE** Discrete Fourier transform, FFT algorithm, no extra workspace, real data values. Double precision version is C06EAF.
- C06FAE** Discrete Fourier transform, FFT algorithm, extra workspace for greater speed, real data values. Double precision version is C06FAF.

PORT subprogram library

- FFTR** Mixed radix fast Fourier transform to find the transform of 2N real data points. Double precision version is DFFTR.
- **FFTRI** Finds the inverse Fourier transform using Fourier coefficients assumed to arise from real data in the time domain. Double precision version is DFFTRI.
- RLTR** An auxiliary routine for use together with FFT to transform 2N real data points. Uses less storage than FFTR. Double precision version is DRLTR.

J1a2 : Complex**CMLIB subprogram library (FFTPKG sublibrary)**

- CFFTB** Backward complex discrete (fast) Fourier transform. Performs Fourier synthesis.
CFFTF Forward complex discrete (fast) Fourier transform. Performs Fourier analysis.

IMSL subprogram library

- FFT2C** Computes the fast Fourier transform of a complex valued sequence of length equal to a power two.
FFTCC Compute the fast Fourier transform of a complex valued sequence.

NAG subprogram library

- C06EBE** Discrete Fourier transform, FFT algorithm, no extra workspace, complex data values (Hermitian sequence). Double precision version is C06EBF.
C06ECE Discrete Fourier transform, FFT algorithm, no extra workspace, complex data values (general sequence). Double precision version is C06ECF.
C06FBE Discrete Fourier transform, FFT algorithm, extra workspace for greater speed, complex data values (Hermitian sequence). Double precision version is C06FBF.
C06FCE Discrete Fourier transform, FFT algorithm, extra workspace for greater speed, complex data values (general sequence). Double precision version is C06FCF.

PORT subprogram library

- FFT** Compute FFT of complex data sequence (forward or inverse) any number of points. Useful for multi-variate transforms. Uses only real arithmetic. Double precision version is DFFT.
• FFTC Mixed radix fast Fourier transform of complex data. Two arrays used for complex data. Double precision version is DFFTC.
• FFTCI Finds the inverse fast Fourier transform, given the Fourier coefficients in the frequency domain. Double precision version is DFFTCI.

J1a3 : Trigonometric (sine, cosine)**CMLIB subprogram library (FFTPKG sublibrary)**

- COSQB** Fast Fourier transform of quarter wave data. Computes a sequence from cosine series representation. Fourier synthesis.
COSQF Computes fast Fourier transform of quarter wave data. Fourier analysis. Computes coefficients in cosine series with odd wave numbers.
COST Computes discrete (fast) cosine transform of even sequence X(I).
SINQB Computes (fast) Fourier transform of quarter wave data. Backward (fast) sine transform. Performs Fourier synthesis.
SINQF Computes (fast) Fourier transform of quarter wave data. Forward (fast) sine transform. Performs Fourier analysis.
SINT Computes (fast) Fourier sine transform of an odd sequence X(I).

IMSL subprogram library

- FFTSC** Compute the sine and cosine transforms of a real valued sequence.

J1b : Multidimensional**IMSL subprogram library**

FFT3D Compute the fast Fourier transform of a complex valued 1,2 or 3 dimensional arra.

NAG subprogram library

C06ADE Discrete Fourier transform, FFT algorithm, complex data values within a multi- variable transform. Double precision version is C06ADF.

PORT subprogram library

FFT Compute FFT of complex data sequence (forward or inverse) any number of points. Useful for multi-variate transforms. Uses only real arithmetic. Double precision version is DFFT.

J2 : Convolutions**IMSL subprogram library**

VCONVO Vector convolution.

NAG subprogram library

C06ACE Circular convolution of two real vectors of period 2^m . Double precision version is C06ACF.

J3 : Laplace transforms**IMSL subprogram library**

FLINV Inverse Laplace transform of a user supplied complex function.

J4 : Hilbert transforms**CMLIB subprogram library (QUADSP sublibrary)**

• **QAWC** Cauchy principal value integrator, using adaptive Clenshaw-Curtis method (real Hilbert transform). Double precision version is DQAWC.

QAWCE Cauchy Principal value integrator, provides more information than QAWC (real Hilbert transform). Double precision version is DQAWCE.

QC25C Uses 25 point Clenshaw-Curtis formula to estimate integral of F times W where $W=1/(X-C)$. Double precision version is DQC25C.

NAG subprogram library

D01AQE Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, weight function $1/(x-c)$ (Hilbert transform). Double precision version is D01AQF.

K: Approximation

Any computer calculation which tries to model a continuous process involves some sort of approximation. In this chapter we consider only those programs which have curve or surface fitting as their ultimate goal. Computer-aided design is an important application of this software. Here one would like a simple mathematical function which represents the shape of an object under study. Such a function might be used, for example, by a numerically controlled milling machine. Another example is in computer graphics, where one wants a visually pleasing curve which approximates given data. Function approximation is a third example. Here one has a complicated mathematical function which must be replaced by a simpler one providing an approximation with a guaranteed maximum error.

The nature of an approximating curve or surface is determined by two fundamental choices: the choices of "norm" and "form". The "norm" is the means for measuring the distance of an approximating function $g(x)$ from the data (x_i, y_i) , $i = 1, \dots, n$. In most cases we want to determine the free parameters of the approximating function so that this distance is minimized. Three important choices are:

- (a) Least squares: minimize $\sum_{i=1}^n (y_i - g(x_i))^2$
- (b) Least absolute value: minimize $\sum_{i=1}^n |y_i - g(x_i)|$
- (c) Minimax: minimize $\max_{1 \leq i \leq n} |y_i - g(x_i)|$

Interpolation is a special case of approximation. Here there are at least as many free parameters in $g(x)$ as there are data points, and we require that $y_i - g(x_i) = 0$ for $i = 1, \dots, n$, that is, $g(x)$ passes through the data points. Interpolation is useful when fitting a curve or surface to data which is known exactly, while least squares is more appropriate for data with inherent error. Software for interpolation is classified in chapter E.

The classical "form" for approximating functions is the polynomial. Unfortunately, polynomials are inadequate for many applications: low degree polynomials lack flexibility, while high degree polynomials can fluctuate wildly, and hence lack physical interpretation. An important breakthrough of the 1960's and 70's were piecewise polynomials (splines). These functions consist of low degree polynomial pieces joined smoothly at a set of breakpoints or knots. Piecewise polynomials are both flexible and stable, and have been used with enormous success by engineers and scientists in areas such as the aerospace and automobile industries for years. Easy to use software for computations with piecewise polynomials is now abundant.

Both polynomials and piecewise polynomials are examples of linear approximating functions. That is, they can be written in the form

$$g(x) = a_1 f_1(x) + a_2 f_2(x) + \dots + a_n f_n(x)$$

where the "basis" functions f_1, f_2, \dots, f_n are fixed and the coefficients a_1, a_2, \dots, a_n are determined from the data. (Spline functions can be written in this form using the so-called B-representation.) Nonlinear approximating functions such as rational functions (the quotient of two polynomials) or sums of exponentials are also often useful, though more difficult to handle computationally.

A wide variety of norms and fitting functions are represented in existing software for approximation. A wealth of software is available for the general least squares approximation problem. Least absolute value and minimax computations are much more difficult and relatively less software is available for these.

Most software for approximation utilizes polynomial or piecewise polynomial fitting functions, although some is also available for trigonometric and rational functions. The ultimate flexibility is offered by routines that work with user-defined fitting functions. In many cases two subprograms are needed to solve an approximation problem: one to determine the coefficients of the approximating function from the data and the second to evaluate the fitted function at one or more points chosen by the user. Occasionally subroutines for evaluating derivatives or definite integrals of the approximating function are also provided.

January 1984

References

1. Barnhill, R. E. and R. F. Riesenfeld (1974). *Computer Aided Geometric Design*, Academic Press, New York.
2. Cheney, E. W. (1966). *Introduction to Approximation Theory*, McGraw-Hill, New York.
3. Davis, P. J. (1975). *Interpolation and Approximation*, Dover Publications, New York.
4. de Boor, C. (1978). *A Practical Guide to Splines*, Springer-Verlag, New York.
5. Rivlin, T. J. (1969). *An Introduction to the Approximation of Functions*, Blaisdell, Waltham, Massachusetts.
6. Schumaker, L. L. (1981). *Spline Functions: Basic Theory*, John Wiley & Sons, New York.

K : Approximation (*search also class L8*)

K1: Least squares approximation

In least squares curve fitting a set of m data values (x_i, y_i) , $i = 1, \dots, m$ are approximated by a function $g(x)$ which depends on n unknown parameters a_1, a_2, \dots, a_n . The set of parameter values is found which minimizes

$$\sum_{i=1}^m w_i (y_i - g(x_i))^2$$

where the w_i are suitably chosen positive weights. In statistics this is known as the regression problem. Software which performs least squares approximation with statistical analysis of the results is found in class L8. This class also overlaps with chapter G (Optimization) where software for "minimizing sums of squares" can be found.

There is an important distinction between linear and nonlinear least squares approximation. In the former the unknown parameters appear as linear coefficients of the known fitting functions, whereas in the latter they may appear in any position. For example, in the fitting function

$$y = ae^{bx} + ce^{dx}$$

a and c are linear parameters while b and d enter nonlinearly. This concept of linearity refers only to the unknown parameters and not to the fitting functions. These functions can be highly nonlinear functions of the x values, as in the example above.

It is also possible to impose side conditions or constraints on least squares fits. For example, one might force the fitted function to pass through a data point, or to have non-negative slope at a given point.

Linear least squares approximation problems have an equivalent matrix formulation, and hence algorithms for these problems are special cases of those for solving overdetermined systems of linear equations (described in class D9). Each of the CMLIB, IMSL, NAG, and PORT libraries have subroutines for linear least squares curve fitting using piecewise polynomials (splines). The NAG library provides subroutines using polynomial basis functions, while the IMSL library provides a subroutine based on user-defined fitting functions. Several subroutines are also available for the linearly constrained case.

In the nonlinear case iterative methods must be employed, and these are special cases of the more general optimization routines described in chapter G. Typically, such algorithms require derivatives of the function with respect to the parameters evaluated at the data points and the current parameter estimates. Some modules allow the user the option of either providing these derivatives or having them estimated by finite differences. Each of the CMLIB, IMSL, MATHWARE and NAG libraries have subprograms for nonlinear least squares problems. In addition, an interactive system called INVAR is available on the Sperry 1100 system, with versions supporting graphical output on line printers or Tektronix terminals (using DISSPLA).

January 1984

References

1. de Boor, C. (1978). *A Practical Guide to Splines*, Springer-Verlag, New York.
2. Lawson, C. L. and R. J. Hanson (1974). *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, New Jersey.
3. Dennis, J. E. and R. B. Schnabel (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, Chapter 10.

K1 : Least squares (L_2) approximation

K1a : Linear least squares (*search also classes D5, D6, D9*)

K1a1 : Unconstrained

K1a1a : Univariate data (curve fitting)

K1a1a1 : Polynomial splines (piecewise polynomials)**CMLIB subprogram library (FC sublibrary)**

FC Fits piecewise polynomial to discrete data with equality and inequality constraints.

IMSL subprogram library

ICSFKU Least squares approximation by cubic splines fixed knots.

ICSSCU Cubic spline data smoother.

• **ICSSCV** Cubic spline data smoother (easy-to-use version).

ICSVKU Least squares approximation by cubic splines - variable knots.

NAG subprogram library

E02BAE Least-squares curve fit by cubic splines (including interpolation). Double precision version is E02BAF.

PORT subprogram library

DL2SF Fits discrete data with a B-spline of order K, by least squares. Double precision version is DDL2SF.

DL2SW Fits discrete data with a B-spline of order k, by weighted least squares. Double precision version is DDL2SW.

L2SFF Obtains a weighted least square expansion of a known function in terms of B-splines of order K, at given mesh points. Double precision version is DL2SFF.

L2SFH Obtains a weighted least square expansion of a known function in and its derivatives in terms of B-splines of order K at given mesh points. Double precision version is DL2SFH.

K1a1a2 : Polynomials**NAG subprogram library**

E02ADE Least-squares curve fit by polynomials, arbitrary data points. Double precision version is E02ADF.

E02AFE Least-squares curve fit by polynomials, special data points (including interpolation). Double precision version is E02AFF.

K1a1a3 : Other functions (e.g., rational, trigonometric, user-specified)**IMSL subprogram library**

IFLSQ Least squares approximation with user supplied functions.

K1a1b : Multivariate data (surface fitting)**NAG subprogram library**

E02CAE Least-squares surface fit by polynomials, for data on lines. Double precision version is E02CAF.

E02DAE Least-squares surface fit by bicubic splines. Double precision version is E02DAF.

K1a2 : Constrained

K1a2a : Linear constraints**CMLIB subprogram library (FC sublibrary)**

- FC** Fits piecewise polynomial to discrete data with equality and inequality constraints.
- LPDP** Solves least projected distance problem.
- **LSEI** Solves linearly constrained least squares problem with equality and inequality constraints. Covariance matrix opt output.
- WNNLS** Solves linearly constrained non-negative least squares problem.

NAG subprogram library

- E02AGE** Least-squares curve fit by polynomials, arbitrary data points, values and derivatives may be constrained. Double precision version is E02AGF.

K1a2b : Nonlinear constraints**K1b :** Nonlinear least squares**K1b1 :** Unconstrained**K1b1a :** Smooth functions**K1b1a1 :** User provides no derivatives**CMLIB subprogram library (NL2SN sublibrary)**

- NL2SN** Minimizes a nonlinear sum of squares using residual values only. Double precision version is DNL2SN.

CMLIB subprogram library (SNLS1E sublibrary)

- SNLS1** Minimizes the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm. Flexible usage, including various options for providing Jacobian. Covariance matrix is available via the subroutine SCOV.
- **SNLS1E** Minimizes the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm. An easy to use driver for SNLS1. The covariance matrix is available by calling the subroutine SCOV.

IMSL subprogram library

- ZXSSQ** Minimum of the sum of squares of m functions in n variables using a finite difference Levenberg-Marquardt algorithm.

INVAR program library

- INVAR1** Interactive program for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results. Line printer graphics only.
- INVAR2** Interactive program for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results and DISPLA graphics.

NAG subprogram library

- E04FCE** Unconstrained minimum, sum of squares, n variables (comprehensive), using function values only, combined Gauss-Newton and modified Newton algorithm. Double precision version is E04FCF.
- **E04FDE** Unconstrained minimum, sum of squares, n variables (easy-to-use), function values only, Gauss-Newton and modified Newton algorithm. Double precision version is E04FDF.

K1b1a2 : User provides first derivatives

CMLIB subprogram library (NL2SN sublibrary)

- NL2S1** Minimizes a nonlinear sum of squares using both residual and gradient values supplied by the user. Double precision version is DNL2S1.

CMLIB subprogram library (SNLS1E sublibrary)

- SNLS1** Minimizes the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm. Flexible usage, including various options for providing Jacobian. Covariance matrix is available via the subroutine SCOV.
- **SNLS1E** Minimizes the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm. An easy to use driver for SNLS1. The covariance matrix is available by calling the subroutine SCOV.

INVAR program library

- INVAR1** Interactive program for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results. Line printer graphics only.
- INVAR2** Interactive program for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results and DISSPLA graphics.

MATHWARE subprogram library (NASHLIB sublibrary)

- A23MRT** Modified Marquardt procedure for minimizing a nonlinear sum of squares function.

NAG subprogram library

- E04GBE** Unconstrained minimum, sum of squares, n variables (comprehensive), using first derivatives, Gauss-Newton and quasi-Newton algorithm. Double precision version is E04GBF.
- **E04GCE** Unconstrained minimum, sum of squares, n variables (easy-to-use), using first derivatives, Gauss-Newton and quasi-Newton algorithm. Double precision version is E04GCF.
- E04GDE** Unconstrained minimum, sum of squares, n variables (comprehensive), using first derivatives, Gauss-Newton and modified Newton algorithm. Double precision version is E04GDF.
- **E04GEE** Unconstrained minimum, sum of squares, n variables (easy-to-use), using first derivatives, Gauss-Newton and modified Newton algorithm. Double precision version is E04GEF.

K1b1a3 : User provides first and second derivatives

NAG subprogram library

- E04HEE** Unconstrained minimum, sum of squares, n variables (comprehensive), using second derivatives, Gauss-Newton and modified Newton algorithm. Double precision version is E04HEF.
- **E04HFE** Unconstrained minimum, sum of squares, n variables (easy-to-use), using second derivatives, Gauss-Newton and modified Newton algorithm. Double precision version is E04HFF.

K1b1b : General functions

K1b2 : Constrained

K1b2a : Linear constraints

K1b2b : Nonlinear constraints

K2 : Minimax (L_∞) approximation

IMSL subprogram library

IRATCU Rational weighted Chebyshev approximation of a continuous function.

RLLMV Perform linear regression using the minimax criterion.

NAG subprogram library

E02ACE Minimax curve fit by polynomials. Double precision version is E02ACF.

E02GCE Calculates an L_∞ solution to an over-determined system of linear equations. Double precision version is E02GCF.

PORT subprogram library

BURAM Finds the best uniform rational approximation to a given function on a specified mesh. Double precision version is DBURAM.

BURM1 Finds the best uniform rational approximation to a given function on a specified mesh, starting from a given initial approximation. Double precision version is DBURM1.

K3 : Least absolute value (L_1) approximation

IMSL subprogram library

RLLAV Perform linear regression using the least absolute values criterion.

NAG subprogram library

E02GAE L_1 -approximation by general linear function. Double precision version is E02GAF.

E02GBE L_1 -approximation by general linear function subject to linear inequality constraints. Double precision version is E02GBF.

K4 : Other analytic approximations (e.g., Taylor polynomial, Pade)

NAG subprogram library

E02RAE Pade-approximants. Double precision version is E02RAF.

K5 : Smoothing

IMSL subprogram library

- ICSMOU One-dimensional data smoothing by error detection.
 ICSSCU Cubic spline data smoother.
 • ICSSCV Cubic spline data smoother (easy-to-use version).

STATLIB subprogram library

- MOVAVG Computes a k-term symmetric moving average of a series.

K6 : Service routines (e.g., mesh generation, evaluation of fitted functions) (*search also class N5*)

CMLIB subprogram library (BSPLINE sublibrary)

- BFQAD** Integrates function times derivative of B-spline from X1 to X2. The B-spline is in "B" representation. Double precision version is DBFQAD.
BSPEV Calculates the value of a spline and its derivatives at X from its "B" representation. Double precision version is DBSPEV.
BSPPP Converts from "B" representation of B-spline to piecewise polynomial representation. Double precision version is DBSPPP.
BSPVD Calculates value and derivatives of order less than NDERIV of all B-spline basis functions which do not vanish at X. Double precision version is DBSPVD.
BSPVN Calculates the value of all (possibly) nonzero B-spline basis functions at X of a given order. Double precision version is DBSPVN.
BSQAD Computes the integral of a B-spline from X1 to X2. The B-spline must be in "B" representation. Double precision version is DBSQAD.
BVALU Calculates (at X) the value of the IDERIV-th derivative of the B-spline from its "B" representation. Double precision version is DBVALU.
INTRV Computes the index into a knot or breakpoint sequence corresponding to a given point X. Double precision version is DINTRV.
PFQAD Computes integral on (X1,X2) of product of function and the ID-th derivative of B-spline which is in piecewise polynomial representation. Double precision version is DPFQAD.
PPQAD Computes the integral of a B-spline from X1 to X2. The B-spline must be in piecewise polynomial representation. Double precision version is DPPQAD.
PPVAL Calculates (at X) the value of the IDERIV-th derivative of the B-spline from its piecewise polynomial representation. Double precision version is DPPVAL.

IMSL subprogram library

- DBCEVL** Bicubic spline mixed partial derivative evaluator.
DCSEVU Cubic spline first and second derivative evaluator.
DCSQDU Cubic spline quadrature.
IBCEVL Evaluation of a bicubic spline.
ICSEVU Evaluation of a cubic spline.

NAG subprogram library

- E02AEE** Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form (simplified parameter list). Double precision version is E02AEF.
E02AHE Derivative of fitted polynomial in Chebyshev series form. Double precision version is E02AHF.
E02AJE Integral of fitted polynomial in Chebyshev series form. Double precision version is E02AJF.
E02AKE Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form. Double precision version is E02AKF.

E02BBE	Evaluation of fitted functions, cubic spline as E02BAE, function only. Double precision version is E02BBF.
E02BCE	Evaluation of fitted functions, cubic spline as E02BAE, function and derivatives. Double precision version is E02BCF.
E02BDE	Evaluation of the definite integral of a cubic spline which was obtained by using E02BAE. Double precision version is E02BDF.
E02CBE	Evaluation of fitted functions, polynomial in two variables as E02CAE. Double precision version is E02CBF.
E02DBE	Evaluation of fitted functions, bicubic spline as E02DAE. Double precision version is E02DBF.
E02RBE	Evaluation of fitted functions, rational function as E02RAE. Double precision version is E02RBF.
E02ZAE	Sort 2-d data into panels for fitting or evaluating bicubic splines. Double precision version is E02ZAF.

PORT subprogram library

BSPL1	Evaluates, at a given set of points in a specified mesh interval, basis splines together with selected orders of derivatives. Double precision version is DBSPL1.
BSPLD	Evaluates at a given set of points in a specified mesh interval, basis splines and their derivatives. Double precision version is DBSPLD.
BSPLI	Obtains the integrals of basis splines, from the left-most mesh point to a specified set of points. Double precision version is DBSPLI.
BSPLN	Evaluates at a given set of points in a specified mesh interval, all the basis splines which are nonzero in that interval. Double precision version is DBSPLN.
CSPFE	Evaluates a cubic spline function which has already been fit to n input data pairs (x,y) by CSPFI. Double precision version is DCSPFE.
DLUMD	Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points.
EEBSF	Estimates the error in a given B-spline fit to a function, f , by refining the mesh. Double precision version is DEEBSF.
EEBSI	Estimates the error in a given B-spline fit to a function f by refining the mesh intervals selected by user. Double precision version is DEEBSI.
EESFF	Finds the maximum absolute error in a given B-spline fit to a function, f . Double precision version is DEESFF.
EESFI	Finds the maximum absolute error in a given B-spline fit to a function, f , on a set of user selected intervals. Double precision version is DEESFI.
IDMNPB	Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each mesh interval.
ILUMB	Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points for B-spline use. Double precision version is IDLUMB.
ILUMD	Given a basic mesh, this subdivides each interval into the same number or uniformly spaced points for B-spline use. Double precision version is IDLUMD.
IMNPB	Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each mesh interval.
IPUMB	Given a basic mesh, this subdivides each interval. Number of points per interval can vary, but uniform in each subdivision. Double precision version is IDPUMB.
IPUMD	Given a basic mesh, this subdivides each interval with a variable number of points. Points are uniform in each interval. Double precision version is IDPUMD.
IUMB	Given interval endpoints, this generates a uniform mesh for B-spline use. Double precision version is IDUMB.
IUMD	Given interval endpoints, this generates a uniform mesh. Double precision version is IDUMD.
LUMB	Given a basic mesh, this subdivides each interval uniformly for B-spline use. Multiplicities are allowed. Double precision version is DLUMB.

LUMD	Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points. Double precision version is IDUMD.
MNPB	Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each interval. Double precision version is DMNPB.
PUMB	Given a basic mesh, this subdivides each interval into a uniform but variable number of points. Multiplicities can occur. Double precision version is DPUMB.
PUMD	Given a basic mesh, this subdivides each interval into a uniform but variable number of points. Double precision version is DPUMD.
SPLN1	Evaluates a function and derivatives described previously by an expansion in terms of B-splines. Double precision version is DSPLN1.
SPLN2	Evaluates a function described by a previously determined expansion in B-splines. More flexible than SPLN1. Double precision version is DSPLN2.
SPLND	Evaluates at a given set of points a function described by a previously determined expansion in terms of B-splines. Double precision version is DSPLND.
SPLNE	Evaluates at a set of points, a function described by a previously determined expansion in terms of B-splines. Double precision version is DSPLNE.
SPLNI	Integrates a function described previously by an expansion in terms of B-splines. Several integrations can be performed in one call. Double precision version is DSPLNI.
UMB	Given interval endpoints, this generates a uniform mesh, with needed multiplicities for B-spline use. Double precision version is DUMB.
UMD	Given interval endpoints, this generates a uniform mesh of distinct points. Double precision version is DUMD.

L: Statistics, Probability

The software cataloged in this chapter covers a wide spectrum of statistical computations—from elementary summary statistics to sophisticated analyses such as regression. Subroutine libraries, program libraries, and interactive systems are each represented. Output ranges from a single number to encyclopedic printouts to graphical displays. The following choices should be considered when selecting statistical software.

1. **Accuracy/precision.** Accuracy of numerical results is crucial. Two different formulas for computing a variance, for example, though mathematically equivalent, may produce different results on computers. Users should select software with the appropriate precision, and that software should report precision problems.
 2. **Documentation.** Computational formulas and procedures should be fully documented.
 3. **Computer.** Because of physical size, libraries are not necessarily available on all machine types (large mainframes, minicomputers, and microcomputers). Proprietary libraries are commonly leased only for specific machines.
 4. **Size of data set.** Whether or not a data set of a particular size can fit on a given computer must be considered when selecting statistical software. Some software is available for analyzing large data sets which cannot be stored in direct-access memory. Programs in a program library and interactive systems may or may not have data set size constraints.
 5. **Data entry.** Since some libraries are unavailable on small computers, and data are sometimes difficult (bearing on impossible) to transport from small to large computers, trouble may arise in bringing software and data together.
 6. **Analyst's time.** A user of statistical software needs to know the language used to communicate with the software. This language usually is Fortran in the case of subroutine libraries, while each stand-alone program library and each interactive system has its own command language. In order to save time, therefore, users are often inclined to use familiar software. Users should not be reluctant to use unfamiliar software, however, because a working knowledge of any software language can generally be quickly acquired. On the other hand, the user is strongly warned against applying the "try everything" approach to the statistical analysis of data, which may be a temptation given the ease of software use.
- The analysis capabilities of any one statistical software library are not entirely duplicated in another, although there is overlap. While the subroutines from several libraries can usually easily be incorporated in a single program, it is more difficult to use several stand-alone programs or interactive systems for one analysis.
7. **Amount of output.** Some statistical software produces encyclopedic printed output—this is especially true of programs run in batch mode. Software designed for interactive use commonly provides relatively terse output, since a screen image provides a natural bound on the amount of output useable at any one time. The form of printed output from subroutines can vary from none (results are returned via the call sequence) to encyclopedic.
 8. **Storage/time trade-off.** Some statistical computations make efficient use of time, others of storage, and commonly not of both simultaneously. This trade-off should be considered when selecting software for an analysis involving a data set of a given size and frequency of execution.
 9. **Graphics.** Researchers are strongly encouraged to plot their data—a picture may reveal useful information not otherwise available through numerical summaries and statistical analyses.

The libraries providing statistical software in this edition of GAMS are:

Subroutine libraries: CMLIB (several sublibraries), DATAPAC, IMSL, NAG, STATLIB
Program libraries: BMDP, INVAR
Interactive systems: Minitab, Spectran

Consult the Library Reference for further information about these libraries.

Each user-callable subroutine in a subroutine library, and each program in a program library, is an entity which is cataloged at one or more classes in the classification scheme, and is an entry in the Module Dictionary. Interactive systems have been handled differently. For the Minitab system, some of the 150 commands are quite elementary, and these commands are grouped under the name "Minitab." For example, there is a "Minitab" entry under class L2a for data transformation which lists approximately two dozen Minitab commands. The whole Spectran system is classified at L10f for spectral analysis.

The following software is scheduled for inclusion in future editions of GAMS:

DATAPLOT (Filliben, 1980), an interactive system for statistical data analysis and Tektronix graphics
GLIM (Baker and Nelder, 1978), an interactive system for analysis of variance and other statistical analyses
OMNITAB (Hogben and Peavy, 1971, 1977), an interactive system for statistical and other types of data analyses
ROSEPACK (Coleman et al., 1980), an interactive system for robust regression
SPSS (Nie et al., 1975; Hull and Nie, 1981), a program for a wide variety of statistical analyses.

January 1984

REFERENCES

1. Anderson, T.W. (1958). *An Introduction to Multivariate Statistical Analysis*, John Wiley & Sons, New York.
2. Baker, R.J. and J.A. Nelder (1978). *The GLIM System Release 3: Generalized Linear Interactive Modeling*, Numerical Algorithms Group, Oxford, 160 pp.
3. Barnett, V., and T. Lewis (1978). *Outliers in Statistical Data*, John Wiley & Sons, New York.
4. Coleman, D. et al. (1980). A System of Subroutines for Iteratively Reweighted Least Squares Computations, *ACM Transactions on Mathematical Software*, Vol. 6, No. 3, pp. 327-336.
5. Daniel, W.W. (1978). *Applied Nonparametric Statistics*, Houghton-Mifflin, New York.
6. Filliben, J.J. (1980). A Review of DATAPLOT - An Interactive High-Level Language for Graphics, Non-Linear Fitting, Data Analysis, and Mathematics, in *Proceedings of the Statistical Computing Section*, American Statistical Association, Washington, D.C.
7. Gnanadesikan, R. (1977). *Methods for Statistical Data Analysis of Multivariate Observations*, John Wiley & Sons, New York.
8. Hogben, D., and S.T. Peavy (1971). *OMNITAB II User's Reference Manual*, NBS Technical Note 552, 254 pp.
9. Hogben, D., and S.T. Peavy (1977). *OMNITAB II User's Reference Manual 1977 Supplement*, NBSIR 77-1276.
10. Hogg, R.V. and A.T. Craig (1978). *Introduction to Mathematical Statistics*, Macmillan, New York.
11. Hollander, M., and D.A. Wolfe (1973). *Nonparametric Statistical Methods*, Wiley-Interscience, New York.
12. Hull, C.H., and N.H. Nie (1981). *SPSS Update 7-9: New Procedures and Facilities for Releases 7-9*, McGraw-Hill, New York.
13. Lehmann, E.L. (1959). *Testing Statistical Hypotheses*, John Wiley & Sons, New York.
14. Natrella, M.G. (1963). *Experimental Statistics*, NBS Handbook 91, U.S. Government Printing Office, Washington, D.C.
15. Nie, N.H., C.H. Hull, J.G. Jenkins, et al. (1975). *SPSS: Statistical Package for the Social Sciences*, McGraw-Hill, New York.
16. Randles, R.H., and D.A. Wolfe (1979). *Introduction to the Theory of Nonparametric Statistics*, Wiley-Interscience, New York.
17. Tukey, J.W. (1977). *Exploratory Data Analysis*, Addison-Wesley, Reading, Massachusetts.

L :	Statistics, probability
------------	-------------------------

L1: Data Summarization

Data summarization is often the first step in the analysis of a data set. It attempts to provide insight into the data through one or a few values which characterize, for example, location or correlation. Classical summary statistics satisfy some optimality criteria when the data are, say, normally distributed. Nonparametric or distribution-free statistics often provide useful information when the distribution of the data is not known.

The software in this chapter is purely for summarizing data, not for elementary statistical analyses such as inference or hypothesis testing (that is in chapter L4), nor for more advanced statistical analyses such as in chapters L7 through L15.

Chapter L1 is first organized by the type of data: one univariate quantitative sample (L1a), one univariate qualitative (proportional) sample (L1c), two or more univariate samples or one multivariate sample (L1e), and two or more multivariate samples (L1f). The labeling of these subchapters parallels that of chapter L4; there is no need for a class for two or more univariate quantitative samples here since software for this task is the same as the software for one multivariate sample, but there is a need to distinguish in chapter L4. There is currently no software in GAMS for summarizing qualitative data; see class L4c for software for summarization and inference. There is also currently no software classified in L1f.

Software is classified both at leaves of the tree and at higher levels; the former compute individual summary statistics while the latter compute a variety of such statistics.

Consult the references listed in the introduction to chapter L for more information about data summarization.

January 1984

L1 :	Data summarization
-------------	--------------------

L1a : One univariate quantitative sample

L1a1 : Ungrouped data

BMDP program library

P1D Prints univariate statistics (mean, standard deviation, std. error of mean, coefficient of variation, extreme values, extreme z-scores, range) for each variable. Options: statistics for each level of each grouping variable, sorting, printing all cases OR only cases with values missing or values outside specified limits.

P2D For each variable, prints frequency and percent for each distinct value; mean, median, mode, standard deviation, std. errors of mean and median, skewness, kurtosis, half interquartile range; histogram, and stem-and-leaf plot. Options: initially round or truncate, three robust location estimates.

DATAPAC subprogram library

LOC Computes 4 estimates (midrange, mean, midmean, and median) of the data in the input vector X.

IMSL subprogram library

BDLTV Produce letter value summary.

BEIUGR Estimation of basic statistical parameters using ungrouped data.

NMTIE Tie statistics, given a sample of observations.

MINITAB interactive system

MINITAB Minitab's vector summarization commands include COUNT, SUM, MEAN, MAX, MIN, MEDIAN, N, NMISS (number of missing values), STDEV, SSQ (sum of squares), DESCRIBE (N, MEAN, MEDIAN, STDEV, MAX, MIN, 5% trimmed mean, quartiles) for columns or rows (use prefix R, e.g., RMEAN) of data in the Minitab worksheet.

NAG subprogram library

G01AAE Simple descriptive statistics, one variable, from raw data. Double precision version is G01AAF.

STATLIB subprogram library

STATS Computes 53 descriptive statistics for a single random sample.

STATSS Computes 53 descriptive statistics for a single random sample with individual weights assigned to each observation and computed results returned to the user.

STATSW Computes 53 descriptive statistics for a single random sample with individual weights assigned to each observation.

L1a1a : Location

DATAPAC subprogram library

MEAN Computes the sample mean of the data in the input vector X.

MEDIAN Computes the sample median of the data in the input vector X.

MIDM Computes the sample midmean, i.e. the sample 25% (on each side) trimmed mean of the data in the input vector X.

TRIM Computes the sample trimmed mean of the data in the input vector X.

WIND Computes the sample Windsorized mean of the data in the input vector X.

L1a1b : Dispersion**DATAPAC subprogram library**

MIDR	Computes the sample midrange of the data in the input vector X.
RANGE	Computes the sample range of the data in the input vector X.
RELSD	Computes the sample relative standard deviation of the data in the input vector X.
SCALE	Computes 4 estimates of the scale (variation, scatter, dispersion) of the data in the input vector X.
SD	Computes the sample standard deviation (with denominator N-1) of the data in the input vector X.
VAR	Computes the sample variance (with denominator N-1) of the data in the input vector X.

L1a1c : Shape**DATAPAC subprogram library**

STMOM3	Computes the sample standardized third central moment of the data in the input vector X.
STMOM4	Computes the sample standardized fourth central moment of the data in the input vector X.

L1a1d : Distribution, density**BMDP program library**

P4D	Counts frequency of each number, letter, or symbol in single-column fields (A1 format). Options: input case label variables in A4 format, diagnostic printing useful in preliminary data screening. Specified characters may be replaced by blanks or symbols.
------------	--

DATAPAC subprogram library

COUNT	Computes the number of observations between XMIN and XMAX (inclusively) in the input vector X.
FREQ	Computes the sample frequency and sample cumulative frequency for the data in the input vector X.
PROPOR	Computes the sample proportion which is the proportion of data between XMIN and XMAX (inclusively) in the input vector X.
SAMPP	Computes the sample 100P percent point (where P is between 0.0 and 1.0, exclusively) of the data in the input vector X.

L1a2 : Ungrouped data with missing values**BMDP program library**

P1D	Prints univariate statistics (mean, standard deviation, std. error of mean, coefficient of variation, extreme values, extreme z-scores, range) for each variable. Options: statistics for each level of each grouping variable, sorting, printing all cases OR only cases with values missing or values outside specified limits.
P2D	For each variable, prints frequency and percent for each distinct value; mean, median, mode, standard deviation, std. errors of mean and median, skewness, kurtosis, half interquartile range; histogram, and stem-and-leaf plot. Options: initially round or truncate, three robust location estimates.

IMSL subprogram library

BESTAT	Computations of basic univariate statistics from data possibly containing missing values, with weighting on option.
---------------	---

MINITAB interactive system

MINITAB Minitab's vector summarization commands include COUNT, SUM, MEAN, MAX, MIN, MEDIAN, N, NMISS (number of missing values), STDEV, SSQ (sum of squares), DESCRIBE (N, MEAN, MEDIAN, STDEV, MAX, MIN, 5% trimmed mean, quartiles) for columns or rows (use prefix R, e.g., RMEAN) of data in the Minitab worksheet.

L1a3 : Grouped data

IMSL subprogram library

BEGRPS Moments estimation for grouped data with and without Sheppards corrections.

BEIGRP Estimation of basic statistical parameters using grouped data.

NAG subprogram library

G01ADE Simple descriptive statistics, one variable, from frequency table. Double precision version is G01ADF.

L1a3a : Location

L1a3b : Dispersion

L1a3c : Shape

L1c : One univariate qualitative (proportional) sample

L1e : Two or more univariate samples or one multivariate sample

L1e1 : Ungrouped data

IMSL subprogram library

BECOR Estimates of means, standard deviations, and correlation coefficients (out-of-core version).

BECORI Estimates of means, standard deviations, and correlation coefficients (in-core version).

BECOVM Means and variance-covariance matrix.

BECVL Variances and covariances of linear functions (out-of-core version).

BECVLI Variances and covariances of linear functions (in-core version).

MINITAB interactive system

TABLE Produces and prints one-way, two-way, and multi-way tables of counts with 20 optional subcommands for summarizing (e.g., cell mean, standard deviation), marginals, performing chi-square tests for each 2-way table, handling missing values, and selecting forms of input and output.

NAG subprogram library

G01ABE Simple descriptive statistics, two variables, from raw data. Double precision version is G01ABF.

G02BAE Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in the array X.

Double precision version is G02BAF.

- G02BDE** Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in the array X. Double precision version is G02BDF.
- G02BGE** Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array X. Double precision version is G02BGF.
- G02BKE** Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array X. Double precision version is G02BKF.

L1e1a : Location

L1e1b : Correlation

DATAPAC subprogram library

- CORR** Computes the sample correlation coefficient between the sets of data in the input vectors X and Y.
- SPCORR** Computes the Spearman rank correlation coefficient between the two sets of data in the input vectors X and Y.

MINITAB interactive system

- CORRELATION** Calculates the Pearson product moment correlation coefficient between two or more pairs of vectors, handles missing values, and optionally saves results.

NAG subprogram library

- G02BNE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X, overwriting X with the ranks of the observations. Double precision version is G02BNF.
- G02BPE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X containing missing values, overwriting X with the ranks of the observations. Double precision version is G02BPF.
- G02BQE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X, preserving X. Double precision version is G02BQF.
- G02BRE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X containing missing values, preserving X. Double precision version is G02BRF.
- G02BSE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X containing missing values, preserving X. Double precision version is G02BSF.

L1e2 : Ungrouped data with missing values

BMDP program library

- P8D** Four methods to compute covariance and correlation matrices when data contain missing values or values out of range. Options: weights, summary statistics, save results, pairwise t-tests based on the pattern of incomplete data.
- PAM** Describes pattern of invalid values (missing or out of range) for multivariate data. Options: weights, grouping, estimates covariance and correlation matrices by one of three methods (including maximum likelihood), replace invalid values using means or one of several regression procedures, plots, save results.

IMSL subprogram library

- BECOVW** Means and variance-covariance or correlation matrix from data possibly containing missing observations, with weighting on option.
- BEMMI** Estimates of means, std. devs., correlation coefficients, and coefficients of skewness and kurtosis from a data matrix containing missing observations (in-core version).
- BEMMO** Estimates of means, std. devs., correlation coefficients, and coefficients of skewness and kurtosis from a data matrix containing missing observations (out-of-core version).

MINITAB interactive system

- CORRELATION** Calculates the Pearson product moment correlation coefficient between two or more pairs of vectors, handles missing values, and optionally saves results.
- TABLE** Produces and prints one-way, two-way, and multi-way tables of counts with 20 optional subcommands for summarizing (e.g., cell mean, standard deviation), marginals, performing chi-square tests for each 2-way table, handling missing values, and selecting forms of input and output.

NAG subprogram library

- G02BBE** Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in the array X containing missing values. Double precision version is G02BBF.
- G02BCE** Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in the array X containing missing values. Double precision version is G02BCF.
- G02BEE** Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in array X containing missing values. Double precision version is G02BEF.
- G02BFE** Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in array X containing missing values. Double precision version is G02BFF.
- G02BHE** Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array X containing missing values. Double precision version is G02BHF.
- G02BJE** Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array X containing missing values. Double precision version is G02BJF.
- G02BLE** Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array X containing missing values. Double precision version is G02BLF.
- G02BME** Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array X containing missing values. Double precision version is G02BMF.

L1e3 :	Grouped data
---------------	--------------

L1f :	Two or more multivariate samples
--------------	----------------------------------

L2: Data Manipulation

This chapter contains software for transforming data (L2a), grouping or tallying data (L2b), sampling data (L2c), and selecting a subset of a data set (L2d). The transformation software handles univariate data (e.g., commonly used transformations), multivariate data, and transformations for use in nonparametric (distribution-free) analyses. Software

for transforming time series is classified at L10a. Software for sorting and ranking is classified at N6. Data creation commands are currently classified at L2.

Consult the references listed in the introduction to chapter L for more information about data manipulation.

January 1984

L2 : Data manipulation (*search also class N*)

MINITAB interactive system

- JOIN** Merges constants and/or vectors into vectors.
- SET** Create a constant vector or a vector of integers in increments of 1 or more or with other patterns.

L2a : Transform (*search also class L10a for time series transformations and class N6 for sorting, ranking*)

BMDP program library

- P15** At each pass through the data, computes univariate statistics (choose means, standard deviations, geometric means, harmonic means, extreme values), and transforms or edits the data using statistics computed in the previous pass. Options: printing, save results.

DATAPAC subprogram library

- DISCR2** Discretizes the data in the vector X into NUMCLA classes.
- DISCR3** Discretizes the data in the vector X into NUMCLA classes.
- DISCRE** Discretizes the data of the vector X according to class width.
- REPLAC** Replaces (with the value XNEW) all observations in the vector X which are inside the interval [XMIN, XMAX].

IMSL subprogram library

- BDTRGI** Transgeneration of the columns of a matrix (in-core version).
- BDTRGO** Transgeneration of the columns of a matrix (out-of-core version).
- RLGQMI** Centering of independent variable settings and generation of centered square and cross product terms - in-core version.
- RLGQMO** Centering of independent variable settings and generation of uncentered square and cross product terms - out-of-core version.
- RLPOL** Generate orthogonal polynomials with the associated constants AA and BB.

MINITAB interactive system

- CENTER** Centers data to mean 0, standard deviation 1. Optionally can select location and scale or minimum and maximum.
- DIFFERENCES** Computes all differences $X - Y$ for each value X in one vector and each value Y in a second vector (useful for nonparametric tests and confidence intervals).
- MINITAB** Minitab's vector transformation commands include ADD, SUBTRACT, MULTIPLY, DIVIDE, RAISE, SIN, COS, TAN, ASIN, ACOS, ATAN, LOGE, LOGTEN, EXPONENTIAL, ANTILOG, ABSOLUTE VALUE, ROUND, SIGNS, SQRT, INDICATOR, RECODE, SUBSTITUTE, CONVERT, PARSUM, PARPRODUCT, and LET (to combine commands).
- NSCORES** Calculates normal scores.
- WALSH** Calculates $(X(i) + X(j))/2$ and stores these average and their indices (useful for nonparametric tests and confidence intervals).

L2b :	Group
-------	-------

IMSL subprogram library

BDCOU1	Tally of observations into a one-way frequency table.
BDCOU2	Tally of observations into a two-way frequency table.
BDTAB	Computations of frequencies of multivariate data.
BDTWT	Computations of a two-way frequency table.
GTDDU	D-square tally.
GTPL	Poker test tally of hand types and statistics.
GTPR	Tally of coordinates of pairs (or lagged pairs) of random numbers.
GTRTN	Tally of number of runs up and down.
GTTRT	Tally for triplets test.

NAG subprogram library

G01AEE	Frequency table from raw data. Double precision version is G01AEF.
--------	--

L2c :	Sample
-------	--------

IMSL library

SSPAND	Simple random sampling with proportion data-inferences regarding the population proportion and total.
SSPBLK	Stratified random sampling with proportion data - inferences regarding the population proportion and total.
SSRAND	Simple random sampling with continuous data - inferences regarding the population mean and total using ratio or regression estimation.
SSRBLK	Stratified random sampling with continuous data-inferences regarding the population mean and total using ratio or regression estimation.
SSSAND	Simple random sampling with continuous data - inferences regarding the population mean and total.
SSSBLK	Stratified random sampling with continuous data - inferences regarding the population mean and total.
SSSCAN	Single stage cluster sampling with continuous data - inferences regarding the population mean and total.
SSSEST	Two-stage sampling with continuous data and equisized primary units - inferences regarding the population mean and total.

L2d :	Subset
-------	--------

DATAPAC library

DELETE	Deletes all observations in the vector X which are inside the interval [XMIN, XMAX].
RETAIN	Retains all observations in the vector X which are inside the interval [XMIN, XMAX].
SUBSE1	Carry over into Y all observations of vector X for which the corresponding elements in vector D are in the interval [DMIN,DMAX].
SUBSE2	Carry over into Y all observations of vector X for which the corresponding elements in vector D1 are in the inclusive interval [D1MIN,D1MAX] and also for which the corresponding elements in D2 are in the interval [D2MIN,D2MAX].
SUBSET	Retain all observations in vector X for which the corresponding elements in vector D are in the interval

[DMIN,DMAX].

MINITAB interactive system

MINITAB Minitab's subsetting commands include PICK, CHOOSE, and OMIT for selecting or deleting entries in a vector in a Minitab worksheet.

NAG subprogram library

G02CEE Service routines for multiple linear regression, select elements from vectors and matrices. Double precision version is G02CEF.

L3: Graphics

The usefulness of graphical analysis of data cannot be overemphasized. Important but perhaps subtle information about data not apparent in numerical output can often be easily discovered through inspection of graphical displays.

This edition of GAMS only includes software for line-printer plots. High-resolution graphics software will appear in this chapter and/or in chapter Q (general-purpose graphics) in future editions.

January 1984

References

1. Schmid, C.F. (1983). *Statistical Graphics: Design Principles and Practices*, Wiley-Interscience, New York.
2. Tufte, E. R. (1983). *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, Connecticut.

L3 : Graphics (*search also class Q*)

L3a : Histograms

DATAPAC subprogram library

HIST Produces 2 histograms (with differing class widths) of the data in the input vector X.

PLOTU Produces 4 plots: data plot (X(I) versus I), autoregression plot (X(I) versus X(I-1)), histogram, and normal probability plot.

IMSL subprogram library

USHHST Print a horizontal histogram.

USHST Print a vertical histogram.

USHST2 Print a vertical histogram, plotting two frequencies with one bar of the histogram.

MINITAB interactive system

HISTOGRAM Prints a histogram of the values in each of one or more vectors, with optional user-specification of the first midpoint and the interval width.

NAG subprogram library

G01AJE Prints a histogram on a character printing device, with user control over size, positioning, and the range of data values included. Double precision version is G01AJF.

STATLIB subprogram library

HISTO Produces a histogram and summary statistics.

HISTOC Produces a histogram and summary statistics, with user control of the number of cells, and of the upper and lower histogram boundaries.

L3b : Distribution functions**BMDP program library**

P5D Prints histograms with frequencies and percentages, normal and detrended normal probability plots, halfnormal plots, cumulative frequency distribution plots, and cumulative histograms for ungrouped data or for grouped data – either separately or combined in one plot. Plot options.

IMSL subprogram library

USPC Print a sample pdf, a theoretical pdf and confidence band information; plot these on option.
USPDF Plot of two sample probability distribution functions against their spectra.
USPLO Printer plot of up to ten functions.
USPLOD Printer plot of up to ten functions.

L3c : Scatter diagrams**BMDP program library**

P6D Bivariate (scatter) plots. Options: several variables, or subsets of one variable (symbols identify group membership), on the same plot; prints correlation and linear regression statistics (line is marked on plot frame); user control for plot size, scales, and symbols.

L3c1 : Y vs. X**DATAPAC subprogram library**

PLOT Yields a one-page printer plot of Y(I) versus X(I).
PLOT6 Yields a one-page printer plot of Y(I) versus X(I) for specified axis limits.
PLOTS Yields a one-page printer plot of Y(I) versus X(I) for a subset of the data.
PLOTST Yields a narrow-width (71-character) of Y(I) versus X(I) for a subset of the data.
PLOTT Yields a narrow-width (71-character) plot of Y(I) versus X(I).

MINITAB interactive system

PLOT Prints a scatter diagram, with optional scale specification.

NAG subprogram library

G01AGE Line printer scatter plot of two variables. Double precision version is G01AGF.

STATLIB subprogram library

PLT Displays a 50x100 character line printer scatter plot.
PLTH Displays a 50x50 character line printer scatter plot.
PLTHL Displays a 50x50 character line printer scatter plot with user control of plot limits.
PLTL Displays a 50x100 character line printer scatter plot with user control of plot limits.

L3c2 : Symbol plots**DATAPAC subprogram library**

PLOT10 Yields a one-page printer plot of Y(I) versus X(I) for a subset of the data, with special plot characters,

and with specified axis limits and labels.

PLOT7	Yields a one-page printer plot of Y(I) versus X(I) with special plot characters and for specified axis limits.
PLOT8	Yields a one-page printer plot of Y(I) versus X(I) with special plot characters for a subset of the data with specified axis limits.
PLOT9	Yields a one-page printer plot of Y(I) versus X(I) with special plot characters and for specified axis limits and axis labels.
PLOT C	Yields a one-page printer plot of Y(I) versus X(I) with special plotting characters.
PLOTCT	Yields a narrow-width (71-character) plot of Y(I) versus X(I) with special plotting characters.
PLOTSC	Yields a one-page printer plot of Y(I) versus X(I) with special characters for a subset of the data.
PLTSCT	Yields a narrow-width (71-character) plot of Y(I) versus X(I) with special plot characters and a subset of the data.

MINITAB interactive system

LPLOT	Prints a letter plot with symbols corresponding to numerical "tag" values. Scale specification is optional.
TPLOT	Prints pseudo three-dimensional plot of y versus x versus z, with symbols indicating the values of z, and with optional scale specification.

STATLIB subprogram library

SPLT	Displays a 50x100 character line printer scatter plot with user control of the plotting symbol used for each point.
SPLTH	Displays a 50x50 character line printer scatter plot with user control of the plotting symbol used for each point.
SPLTHL	Displays a 50x50 character line printer scatter plot with user control of the plot limits and of the plotting symbol used for each point.
SPLTL	Displays a 50x100 character line printer scatter plot with user control of the plot limits and of the plotting symbol used for each point.

L3c3 : Multiple plots

MINITAB interactive system

MPLOT	Prints multiple scatter diagrams on the same axis.
--------------	--

STATLIB subprogram library

MPLT	Displays a 50x100 character line printer plot of several dependent variables vs. a common independent variable.
MPLTH	Displays a 50x50 character line printer plot of several dependent variables vs. a common independent variable.
MPLTHL	Displays a 50x50 character line printer plot of several dependent variables vs. a common independent variable, with user control of the plot limits.
MPLTL	Displays a 50x100 character line printer plot of several dependent variables vs. a common independent variable, with user control of the plot limits.

L3c4 : Probability plots

IMSL subprogram library

USPRP	Probability plot.
--------------	-------------------

L3c4b : Beta, binomial

L3c4c : Cauchy, chi-squared

DATAPAC subprogram library

CAUPLT Generates a Cauchy probability plot with median = 0 and 75% point = 1.

CHSPLT Generates a chi-squared probability plot with integer degrees of freedom parameter value = NU.

L3c4d : Double exponential

DATAPAC subprogram library

DEXPLT Generates a double exponential (Laplace) probability plot with mean = 0 and standard deviation = sqrt(2).

L3c4e : Exponential, extreme value

DATAPAC subprogram library

EV1PLT Generates an extreme value type 1 probability plot with mean = Euler's number = 0.57721566 and standard deviation = pi/sqrt(6).

EV2PLT Generates an extreme value type 2 probability plot with tail length parameter = GAMMA.

EXPPLT Generates an exponential probability plot with mean = 1 and standard deviation = 1.

L3c4f : F distribution

L3c4g : Gamma, geometric

DATAPAC subprogram library

GAMPLT Generates a gamma probability plot with tail length parameter = GAMMA, mean = GAMMA, and standard deviation = sqrt(GAMMA).

GEOPLT Generates a geometric probability plot with parameter P.

L3c4h : Halfnormal

BMDP program library

P5D Prints histograms with frequencies and percentages, normal and detrended normal probability plots, halfnormal plots, cumulative frequency distribution plots, and cumulative histograms for ungrouped data or for grouped data - either separately or combined in one plot. Plot options.

DATAPAC subprogram library

HFNPLT Generates a halfnormal probability plot with mean = sqrt(2/pi) and standard deviation = 1.

L3c4l : Lambda, logistic, lognormal

DATAPAC subprogram library

LAMPLT Generates a (Tukey) lambda distribution probability plot with tail length parameter ALAMBA.
LGNPLT Generates a lognormal probability plot with mean = \sqrt{e} .
LOGPLT Generates a logistic probability plot with mean = 0 and standard deviation = $\pi/\sqrt{3}$.

L3c4n : Negative binomial, normal

BMDP program library

P5D Prints histograms with frequencies and percentages, normal and detrended normal probability plots, halfnormal plots, cumulative frequency distribution plots, and cumulative histograms for ungrouped data or for grouped data – either separately or combined in one plot. Plot options.

DATAPAC subprogram library

NORPLT Generates a normal (Gaussian) probability plot with mean = 0 and standard deviation = 1.
PLOTU Produces 4 plots: data plot (X(I) versus I), autoregression plot (X(I) versus X(I-1)), histogram, and normal probability plot.

NAG subprogram library

G01AHE Line printer scatter plot of one variable against normal scores. Double precision version is G01AHF.

L3c4p : Pareto, Poisson

DATAPAC subprogram library

PARPLT Generates a Pareto probability plot with tail length parameter GAMMA.
POIPLT Generates a Poisson probability plot with tail length parameter ALAMBA, mean = ALAMBA and standard deviation = $\sqrt{\text{ALAMBA}}$.

L3c4t : t distribution

DATAPAC subprogram library

TPLT Generates a Student's t probability plot with degrees of freedom parameter NU.

L3c4u : Uniform

DATAPAC subprogram library

UNIPLT Generates a uniform probability plot on the unit interval (0,1) with mean = 0.5 and standard deviation = $\sqrt{1/12}$.

L3c4w : Weibull

DATAPAC subprogram library

WEIPLT Generates a Weibull probability plot with tail length parameter GAMMA.

L3c5 : Time series plots (X(i) vs. i, vertical, lag)

DATAPAC subprogram library

PLOTU Produces 4 plots: data plot (X(I) versus I), autoregression plot (X(I) versus X(I-1)), histogram, and normal probability plot.

PLOTX Yields a one-page printer plot of X(I) versus I.

PLOTXT Yields a narrow-width (71-character) plot of X(I) versus I.

PLOTXX Yields a one-page printer plot of X(I) versus X(I-1) for testing autocorrelation.

PLTXXT Yields a narrow-width (71-character) plot of X(I) versus X(I-1) for testing autocorrelation.

MINITAB interactive system

TSPLOT Prints a scatter diagram of a time series, optionally using symbols modulo the period. Handles missing values.

STATLIB subprogram library

VPLT Displays an Nx100 character line printer plot of the N values of a series (horizontal axis) vs. their indices (vertical axis).

VPLT2 Displays an Nx100 character line printer plot of the N values of each of two series (horizontal axis) vs. their indices (vertical axis).

VPLT2L Displays an Nx100 character line printer plot of the N values of each of two series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits.

VPLTB Displays an Nx100 character line printer bar plot of the N values of a series (horizontal axis) vs. their indices (vertical axis).

VPLTBL Displays an Nx100 character line printer bar plot of the N values of a series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits.

VPLTL Displays an Nx100 character line printer plot of the N values of a series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits.

L3d : EDA graphics

IMSL subprogram library

USBOX Print a boxplot (k samples).

USSLF Print a stem-and-leaf display.

MINITAB interactive system

BOXPLOT Prints boxplots - median, hinges, inner and outer fences - for one or more levels. Options: form of plots, notches (confidence interval for population medians).

CPLOT Prints a scatter diagram which condenses as many as 10 lines of plot into one line and trims extreme x- and y-values. Option: form of output.

CTABLE Prints a coded two-way table, each cell of which is coded with one character for features MAXIMUM, MINIMUM, or EXTREME, and codes for values between the hinges, between hinges and inner fences, between inner and outer fences, and beyond the outer fences.

LVALS Prints letter-value display - median, hinges, eighths, etc., and optionally saves results.

ROOTOGRAM Prints a suspended rootogram, i.e. a histogram which has been fit with a Gaussian distribution

based on square roots of the counts of data values in each bin and which uses medians and hinges.
Options: specify Gaussian mean and standard deviation, save results.

STEM-&-LEAF Prints stem-and-leaf display(s), optionally with no trimming of outliers.

L4: Elementary Statistical Inference, Hypothesis Testing

The distinction between this chapter and chapter L1 (for data summarization) is that while the same statistics are often computed both for summarization and for inference and testing, this chapter has the additional feature that assumptions are made about the distribution of the data, and either those assumptions are tested or inferences are made based on the assumptions. Software for more sophisticated statistical analyses are classified in chapters L7 through L15.

The organization of this chapter parallels that of chapter L1. Thus, the software is first distinguished by the number of populations and the dimension of the data (e.g., one univariate quantitative sample (L4a)).

Chapter L4a1a contains software which does both sampling and inference - it may be appropriate for analysis of large data sets. Software for analyzing two or more univariate samples, some of which may be quantitative and other qualitative, is classified at L4d. Software for correlation analysis is classified both at L4e (one multivariate sample) and L11 (correlation analysis). Software is classified both at leaves of the tree and at higher levels in this chapter; the latter perform a variety of testing and inference calculations.

Two general maximum likelihood parameter estimation programs are P3R and PAR from BMDP.

Consult the references listed in the introduction to chapter L for more information about elementary inference and hypothesis testing.

January 1984

L4 :	Elementary statistical inference, hypothesis testing
-------------	--

L4a :	One univariate quantitative sample
--------------	------------------------------------

BMDP program library

P3D	One-sample t-test to test if one group mean is zero (e.g., matched pairs); two-sample t test with and without equal variances assumption, Levene's test for equal variances, histograms. Options: trimmed t test, Hotelling's T-squared and Mahalanobis' D-squared, within-group correlations, data listing.
------------	--

L4a1 :	Ungrouped data
---------------	----------------

L4a1a :	Parameter estimation
----------------	----------------------

DATAPAC subprogram library

TAIL	Performs a symmetric distribution tail length analysis on the data in the input vector X.
-------------	---

IMSL library

SSRAND	Simple random sampling with continuous data - inferences regarding the population mean and total using ratio or regression estimation.
---------------	--

SSRBLK	Stratified random sampling with continuous data-inferences regarding the population mean and total using ratio or regression estimation.
---------------	--

SSSAND	Simple random sampling with continuous data - inferences regarding the population mean and total.
---------------	---

SSSBLK	Stratified random sampling with continuous data - inferences regarding the population mean and total.
---------------	---

SSSCAN	Single stage cluster sampling with continuous data - inferences regarding the population mean and total.
---------------	--

SSSEST Two-stage sampling with continuous data and equisized primary units - inferences regarding the population mean and total.

L4a1a2 : Binomial

IMSL subprogram library

BELBIN Interval estimate of the parameter p of the binomial distribution.

L4a1a5 : Extreme value

DATAPAC subprogram library

EXTREM Performs an extreme value analysis on the data in the input vector X .

L4a1a14 : Normal

DATAPAC subprogram library

NOROUT Performs a normal outlier analysis on the data in the input vector X .

IMSL subprogram library

BEMNON Location (mean) inferences using a sample from a normal population with known variance.

BEMSON Mean and variance inferences using a sample from a normal population.

BENSON Variance inferences using a sample from a normal population with known mean.

GTNOR Test for normality of random deviates.

MINITAB interactive system

TINTERVAL Calculates a t -confidence interval with specified percent confidence.

TTEST Performs one- or two-sided t -tests.

ZINTERVAL Calculates a z -confidence interval with specified percent confidence and standard deviation.

ZTEST Performs a one- or two-sided z -test for a specified standard deviation.

L4a1a16 : Poisson

IMSL subprogram library

BELPOS Interval estimate of the parameter λ of the Poisson distribution.

L4a1a21 : Uniform

IMSL subprogram library

GTMNT Moments and standardized moments of uniform random numbers.

L4a1a23 : Weibull**DATAPAC subprogram library**

WEIB Performs a Weibull distribution analysis on the data in the input vector X.

L4a1b : Distribution-free (nonparametric) analysis**BMDP program library**

P3S Computes and prints results from one or more of the following: sign test, Wilcoxon signed-rank test, Mann-Whitney rank-sum test, Kruskal-Wallis one-way ANOVA, Friedman two-way ANOVA, Kendall's coefficient of concordance, Kendall and Spearman rank-correlation coefficients.

IMSL subprogram library

NBSIGN Sign test (for percentiles).

MINITAB interactive system

WINTERVAL Calculates a one-sample Wilcoxon rank estimate and confidence interval for the center of a symmetric distribution.

WTEST Performs one-sample one- or two-sided Wilcoxon signed-rank tests.

L4a1c : Goodness-of-fit tests**IMSL subprogram library**

GFIT Chi-squared goodness of fit test.

GTCN Sample size or number of class intervals determination for chi-squared test applications.

GTD2T The d-square test.

GTPOK The poker test.

NKS1 Kolmogorov-Smirnov one-sample test.

NAG subprogram library

G08CAE Kolmogorov-Smirnov one-sample distribution test. Double precision version is G08CAF.

L4a1d : Tests on sequences of numbers**DATAPAC subprogram library**

RUNS Performs a run analysis of the data in the input vector X.

IMSL subprogram library

GTPST Pairs test or Goods serial test.

GTRN Runs test.

GTTT Triplets test.

NBCYC Noethers test for cyclical trend.

NBSDI Cox and Stuart sign test for trends in dispersion and location.

MINITAB interactive system

RUNS Performs a two-sided runs test.

L4a1c : Density and distribution function estimation

IMSL subprogram library

NDKER Nonparametric probability density function (one dimensional) estimation by the kernel method.
NDMPLE Nonparametric probability density function (one dimensional) estimation by the penalized likelihood method.
USPC Print a sample pdf, a theoretical pdf and confidence band information; plot these on option.

L4a1f : Tolerance limits

DATAPAC subprogram library

TOL Computes normal and distribution-free tolerance limits for the data in the input vector X.

L4a2 : Ungrouped data with missing values

IMSL subprogram library

BESTA2 Computations of confidence intervals and other basic statistics using output from IMSL routine BESTAT.

MINITAB interactive system

TINTERVAL Calculates a t-confidence interval with specified percent confidence.
TTEST Performs one- or two-sided t-tests.
ZINTERVAL Calculates a z-confidence interval with specified percent confidence and standard deviation.
ZTEST Performs a one- or two-sided z-test for a specified standard deviation.

L4a3 : Grouped data

L4a3a : Parameter estimation

L4a3a14 : Normal

IMSL subprogram library

OTMLNR Maximum likelihood estimation from grouped and/or censored normal data.

L4b : Two or more univariate quantitative samples

BMDP program library

P3D One-sample t-test to test if one group mean is zero (e.g., matched pairs); two-sample t test with and without equal variances assumption, Levene's test for equal variances, histograms. Options: trimmed t

test, Hotelling's T-squared and Mahalanobis' D-squared, within-group correlations, data listing.

L4b1 : Ungrouped data

L4b1a : Parameter estimation

L4b1a14 : Normal

IMSL subprogram library

BECTR Tetrachoric correlation coefficient estimation.

BEPAT Mean and variance inferences using samples from each of two normal populations with unequal variances.

BEPET Mean and variance inferences using samples from each of two normal populations with equal variances.

MINITAB interactive system

TWOSAMPLE Performs a one- or two-sided two-sample t-test of equality of two population means with variance estimated either from each sample or the pooled sample.

TWOT Performs a one- or two-sided two-sample t-test of equality of two population means with variance estimated either from each sample or the pooled sample.

L4b1b : Distribution-free (nonparametric) analysis

BMDP program library

P3S Computes and prints results from one or more of the following: sign test, Wilcoxon signed-rank test, Mann-Whitney rank-sum test, Kruskal-Wallis one-way ANOVA, Friedman two-way ANOVA, Kendall's coefficient of concordance, Kendall and Spearman rank-correlation coefficients.

DATAPAC subprogram library

SPCORR Computes the Spearman rank correlation coefficient between the two sets of data in the input vectors X and Y.

IMSL subprogram library

NAK1 Kruskal-Wallis test for identical populations.

NBQT Cochran q test.

NHINC Includance test.

NKS2 Kolmogorov-Smirnov two-sample test.

NMCC Calculate and test the significance of the Kendall coefficient of concordance.

NMKN Kendall's test for correlation (rank correlation coefficient).

NMKSF Frequency distribution of K and the probability of equalling or exceeding K, where K, the total score from the Kendall rank correlation coefficient calculations, and N, the sample size, are given.

NMKTS K-sample trends test against ordered alternatives.

NRBHA Bhapkar v test.

NRWMD Wilcoxon signed rank test.

NRWRST Wilcoxon rank-sum test.

MINITAB interactive system

KRUSKAL-WALL Perform Kruskal-Wallis test, based on ranks, of the null hypothesis that there is no difference among K population locations against the alternative of at least one difference. (This is a K-sample generalization of the Mann-Whitney-Wilcoxon test and is a nonparametric alternative to one-way ANOVA.).

MANN-WHITNEY Performs one- or two-sided two-sample rank test (a.k.a. Wilcoxon rank test) for the difference between two population medians, and calculates the corresponding point and confidence interval estimates.

NAG subprogram library

G08AAE Sign test on two paired samples. Double precision version is G08AAF.

G08ABE Wilcoxon matched pairs signed ranks test on two paired samples. Double precision version is G08ABF.

G08ACE Median test on two samples of unequal size. Double precision version is G08ACF.

G08ADE Mann-Whitney U-test on two samples of unequal size. Double precision version is G08ADF.

G08BAE Mood's and David's tests on two samples of unequal size. Double precision version is G08BAF.

G08DAE Kendall's coefficient of concordance. Double precision version is G08DAF.

L4b2 : Ungrouped data with missing values

MINITAB interactive system

TWOSAMPLE Performs a one- or two-sided two-sample t-test of equality of two population means with variance estimated either from each sample or the pooled sample.

TWOT Performs a one- or two-sided two-sample t-test of equality of two population means with variance estimated either from each sample or the pooled sample.

L4b3 : Grouped data

L4c : One univariate qualitative (proportional) sample

IMSL subprogram library

SSPAND Simple random sampling with proportion data - inferences regarding the population proportion and total.

SSPBLK Stratified random sampling with proportion data - inferences regarding the population proportion and total.

L4d : Two or more univariate samples

IMSL subprogram library

BESRB Biserial and point-biserial correlation coefficients for a qualitatively dichotomized variable and a numerically measurable and classified variable.

BESRN Biserial correlation coefficient for a qualitatively dichotomized variable and a numerically or qualitatively classified variable.

L4e : One multivariate sample

L4e1 : Ungrouped data

L4e1a : Parameter estimation

L4e1a14 : Normal

STATLIB subprogram library

COREL Performs correlation analysis of a multivariate random sample.

CORELS Performs correlation analysis of a multivariate random sample with computed results returned to the user.

L4e1b : Distribution-free (nonparametric) analysis

L4e2 : Ungrouped data with missing values

L4e2a : Parameter estimation

L4e2b : Distribution-free (nonparametric) analysis

L4e3 : Grouped data

L4e3a : Parameter estimation

L4e3a14 : Normal

IMSL subprogram library

CBNRHO Estimation of the bivariate normal correlation coefficient using a contingency table.

L4e3b : Distribution-free (nonparametric) analysis

L4e4 : Two or more multivariate samples

L4e4a : Parameter estimation

L4e4a14 : Normal

IMSL subprogram library

OIND Wilks test for the independence of k sets of multi-normal variates.

L5: Function Evaluation

Evaluation of probability density and distribution functions and their inverses comprises a major component of special function evaluation. A discussion of the computational aspects of the evaluation of these functions can be found in the introduction to chapter C (Special Functions).

This chapter first distinguishes between univariate and multivariate probability functions, and then between evaluating distribution and density functions and evaluating percent point functions (the inverses of distribution functions) and sparsity functions (the inverses of density functions). The subchapters are then further broken down by name of distribution (in alphabetical order). No distinction is made in the classification scheme between density and distribution functions because there are at present few enough codes in each class.

The cataloged software provides capabilities for the evaluation of a wide variety of univariate distribution and density functions and their inverses; there is little software classified for multivariate functions.

January 1984

L5 : Function evaluation (*search also class C*)

L5a : Univariate

L5a1 : Cumulative distribution functions, probability density functions

L5a1b : Beta, binomial

DATAPAC subprogram library

BINCDF Computes the cumulative distribution function value at X for the binomial distribution with parameters P and N.

IMSL subprogram library

GTPKP Probability distribution of n elements into two equi-probable states.

MDBETA Beta probability distribution function.

MDBIN Binomial probability distribution function.

MINITAB interactive system

BINOMIAL Prints table of binomial probabilities and cumulative distribution function, and optionally saves results.

NAG subprogram library

G01BDE Beta distribution of first kind. Double precision version is G01BDF.

L5a1c : Cauchy, chi-squared

DATAPAC subprogram library

CAUCDF Computes the cumulative distribution function value for the Cauchy distribution with median = 0

and 75% point = 1.

CAUPDF Computes the probability density function value for the Cauchy distribution with median = 0 and 75% point = 1.

CHSCDF Computes the cumulative distribution function value for the chi-squared distribution with degrees of freedom parameter = NU.

IMSL subprogram library

MDCH Chi-squared probability distribution function.

MDCHN Non-central chi-squared probability distribution function.

NAG subprogram library

G01BCE Chi-square distribution. Double precision version is G01BCF.

L5a1d : Double exponential

DATAPAC subprogram library

DEXCDF Computes the cumulative distribution function value for the double exponential (Laplace) distribution with mean = 0.

DEXPDF Computes the probability density function value for the double exponential (Laplace) distribution with mean = 0.

L5a1e : Error function, exponential, extreme value

CMLIB subprogram library (FNLIB sublibrary)

ERF Error function, $= (2 / \text{square root of pi}) * \text{the integral from 0 to x of } e^{-(t^2)} dt$. Double precision version is DERF.

ERFC Complementary error function, $= (2 / \text{square root of pi}) * \text{the integral from x to infinity of } e^{-(t^2)} dt$. Double precision version is DERFC.

DATAPAC subprogram library

EV1CDF Computes the cumulative distribution function value for the extreme value type 1 distribution.

EV2CDF Computes the cumulative distribution function value for the extreme value type 2 distribution with tail length parameter=GAMMA.

EXPCDF Computes the cumulative distribution function value for the exponential distribution with mean = 1 and standard deviation = 1.

EXPPDF Computes the probability density function value for the exponential distribution with mean = 1 and standard deviation = 1.

IMSL subprogram library

ERF Evaluate the error function. Note: the Fortran mathematical subroutine libraries may also contain ERF.

ERFC Evaluate the complemented error function.

MATHWARE subprogram library (STEGUN sublibrary)

ERRINT Computes error function and complementary error function to maximum machine accuracy. To change computers change one line.

NAG subprogram library

- S15ADE** Complement of error function, $\text{erfc}(x)$. Double precision version is S15ADF.
S15AEE Error function, $\text{erf}(x)$. Double precision version is S15AEF.

L5alf : F distribution

DATAPAC subprogram library

- FCDF** Computes the cumulative distribution function value for the F-distribution with degrees of freedom parameters NU1 and NU2.

IMSL subprogram library

- MDFD** F probability distribution function.
MDFDRE F probability distribution function (integer or fractional degrees of freedom).

NAG subprogram library

- G01BBE** F (variance ratio) distribution. Double precision version is G01BBF.

L5alg : Gamma, general, geometric

DATAPAC subprogram library

- GAMCDF** Computes the cumulative distribution function value for the gamma distribution with tail length parameter = GAMMA.
GEOCDF Computes the geometric cumulative distribution function value at the value X with parameter = P.

IMSL subprogram library

- MDGAM** Gamma probability distribution function.
MDGC General cumulative probability distribution function, given ordinates of the density.
NDEST Evaluate probability density function at specified points.

L5alh : Halfnormal, hypergeometric

DATAPAC subprogram library

- HFNCDF** Computes the cumulative distribution function value for the halfnormal distribution with mean = $\sqrt{2/\pi}$ and standard deviation = 1.

IMSL subprogram library

- MDHYP** Hypergeometric probability distribution function.

L5alk : Kolmogorov-Smirnov

IMSL subprogram library

- MDSMR** Kolmogorov-Smirnov statistics asymptotic probability distribution function.

L5a1l : Lambda, logistic, lognormal

DATAPAC subprogram library

- LAMCDF** Computes the cumulative distribution function value for the (Tukey) lambda distribution with tail length parameter = ALAMBA.
- LAMPDF** Computes the probability density function value for the (Tukey) lambda distribution with tail length parameter ALAMBA.
- LGNCDF** Computes the cumulative distribution function value for the lognormal distribution with mean \sqrt{e} .
- LOGCDF** Computes the cumulative distribution function value for the logistic distribution with mean = 0 and standard deviation = $\pi/\sqrt{3}$.
- LOGPDF** Computes the probability density function value for the logistic distribution with mean = 0 and standard deviation = $\pi/\sqrt{3}$.

L5a1n : Negative binomial, normal

DATAPAC subprogram library

- NBCDF** Computes the cumulative distribution function value at X for the negative binomial distribution with parameters P and N.
- NORCDF** Computes the cumulative distribution function value for the normal (Gaussian) distribution with mean = 0 and standard deviation = 1.
- NORPDF** Computes the probability density function value for the normal (Gaussian) distribution with mean = 0 and standard deviation = 1.

IMSL subprogram library

- MDNOR** Normal or Gaussian probability distribution function.
- MSMRAT** Ratio of the ordinate to the upper tail area of the standardized normal (Gaussian) distribution.

NAG subprogram library

- S15ABE** Cumulative normal distribution function, $p(x)$. Double precision version is S15ABF.
- S15ACE** Complement of cumulative normal distribution function, $q(x)$. Double precision version is S15ACF.

L5a1p : Pareto, Poisson

DATAPAC subprogram library

- PARCDF** Computes the cumulative distribution function value for the Pareto distribution with tail length parameter GAMMA.
- POICDF** Computes the cumulative distribution function value at X for the Poisson distribution with tail length parameter ALAMBA.

IMSL subprogram library

- MDTPS** Cumulative probability and, optionally, individual terms of the Poisson probability distribution function.

MINITAB interactive system

- POISSON** Prints table of Poisson probabilities and cumulative distribution function.

L5alt : t distribution**DATAPAC subprogram library**

TCDF Computes the cumulative distribution function value for Student's t distribution with degrees of freedom parameter NU.

IMSL subprogram library

MDTD Students t probability distribution function.

MDTN Non-central t probability distribution function.

MDTNF Integral related to calculation of noncentral t and bivariate normal probability distribution functions.

NAG subprogram library

G01BAE Student's t distribution. Double precision version is G01BAF.

L5alu : Uniform**DATAPAC subprogram library**

UNICDF Computes the cumulative distribution function value for the uniform (rectangular) distribution on the unit interval (0,1).

UNIPDF Computes the probability density function value for the uniform (rectangular) distribution on the unit interval (0,1).

L5alw : Weibull**DATAPAC subprogram library**

WEICDF Computes the cumulative distribution function value for the Weibull distribution with tail length parameter GAMMA.

L5a2 : Inverse cumulative distribution functions, sparsity functions**L5a2b :** Beta, binomial**DATAPAC subprogram library**

BINPPF Computes the percent point function value at P for the binomial distribution with parameters PPAR and N.

IMSL subprogram library

MDBETI Inverse beta probability distribution function.

NAG subprogram library

G01CDE Inverse beta distribution of first kind. Double precision version is G01CDF.

L5a2c : Cauchy, chi-squared

DATAPAC subprogram library

- CAUPPF** Computes the percent point function value for the Cauchy distribution with median = 0 and 75% point = 1.
- CAUSF** Computes the sparsity function value for the Cauchy distribution with median = 0 and 75% point = 1.
- CHSPPF** Computes the percent point function value for the chi-squared distribution with integer degrees of freedom parameter = NU.

IMSL subprogram library

- MDCHI** Inverse chi-squared probability distribution function.

NAG subprogram library

- G01CCE** Inverse chi-square distribution. Double precision version is G01CCF.

L5a2d : Double exponential

DATAPAC subprogram library

- DEXPPF** Computes the percent point function value for the double exponential (Laplace) distribution with mean = 0.
- DEXSF** Computes the sparsity function value for the double exponential (Laplace) distribution with mean = 0 and standard deviation = sqrt(2).

L5a2e : Exponential, extreme value

DATAPAC subprogram library

- EV1PPF** Computes the percent point function value for the extreme value type 1 distribution with mean = Euler's number = 0.57721566.
- EV2PPF** Computes the percent point function value for the extreme value type 2 distribution with tail length parameter = GAMMA.
- EXPPPF** Computes the percent point function value for the exponential distribution with mean = 1 and standard deviation = 1.
- EXPSF** Computes the sparsity function value for the exponential distribution with mean = 1 and standard deviation = 1.

IMSL subprogram library

- MERFCI** Inverse complemented error function.
- MERFI** Inverse error function.

L5a2f : F distribution

IMSL subprogram library

- MDFI** Inverse F probability distribution function.

NAG subprogram library

G01CBE Inverse F (variance ratio) distribution. Double precision version is G01CBF.

L5a2g : Gamma, general, geometric

DATAPAC subprogram library

GAMPPF Computes the percent point function value for the gamma distribution with mean = GAMMA and standard deviation = sqrt(GAMMA).

GEOPPF Computes the percent point function value for the geometric distribution with parameter PPAR.

IMSL subprogram library

MDGCI Inverse of a general cumulative probability distribution function, given ordinates of the density.

L5a2h : Halfnormal

DATAPAC subprogram library

HFNPPF Computes the percent point function value for the halfnormal distribution with mean = sqrt(2/pi) and standard deviation = 1.

L5a2l : Lambda, logistic, lognormal

DATAPAC subprogram library

LAMPPF Computes the percent point function value for the (Tukey) lambda distribution with tail length parameter ALAMBA.

LAMSF Computes the sparsity function value for the (Tukey) lambda distribution with tail length parameter ALAMBA.

LGNPPF Computes the percent point function value for the lognormal distribution with mean = sqrt(e).

LOGPPF Computes the percent point function value for the logistic distribution with mean = 0 and standard deviation = pi/sqrt(3).

LOGSF Computes the sparsity function value for the logistic distribution with mean = 0 and standard deviation = pi/sqrt(3).

L5a2n : Negative binomial, normal, normal scores

DATAPAC subprogram library

NBPPF Computes the percent point function value at P for the negative binomial distribution with parameters PPAR and N.

NORPPF Computes the percent point function value for the normal (Gaussian) distribution with mean = 0 and standard deviation = 1.

NORSF Computes the sparsity function value for the normal (Gaussian) distribution with mean = 0 and standard deviation = 1.

IMSL subprogram library

MDNRIS Inverse standard normal (Gaussian) probability distribution function.

MSENO Expected values of normal order statistics.

NAG subprogram library

- G01CEE** Inverse normal distribution. Double precision version is G01CEF.
G01DAE Calculation of normal scores. Double precision version is G01DAF.

L5a2p : Pareto, Poisson

DATAPAC subprogram library

- PARPPF** Computes the percent point function value for the Pareto distribution with tail length parameter GAMMA.
POIPPF Computes the percent point function value at P for the Poisson distribution with mean = ALAMBA and standard deviation = sqrt(ALAMBA).

L5a2t : t distribution

DATAPAC subprogram library

- TPPF** Computes the percent point function value for the Student's t distribution with degrees of freedom parameter NU.

IMSL subprogram library

- MDSTI** Inverse of a modification of Student's t probability distribution function.

NAG subprogram library

- G01CAE** Inverse Student's t distribution. Double precision version is G01CAF.

L5a2u : Uniform

DATAPAC subprogram library

- UNIPPF** Computes the percent point function value for the uniform (rectangular) distribution on the unit interval (0,1).
UNISF Computes the sparsity function value for the uniform (rectangular) distribution on the unit interval (0,1).

L5a2w : Weibull

DATAPAC subprogram library

- WEIPPF** Computes the percent point function value for the Weibull distribution with tail length parameter GAMMA.

L5b : Multivariate

L5b1 : Cumulative distribution functions, probability density functions

L5b1n : Normal

IMSL subprogram library

MDBNOR Bivariate normal probability distribution function.

MDTNF Integral related to calculation of noncentral t and bivariate normal probability distribution functions.

L6: Pseudo-Random Number Generation

The generation of pseudo-random numbers according to certain statistical distributions is an area of statistical computing which has important applications in the design of experiments, the development and analysis of statistical methods, the analysis of algorithms, and simulation studies.

Generation of uniform pseudo-random numbers is central, primarily because pseudo-random numbers from other distributions can often be obtained by transformation of uniform numbers. Efficient methods have been developed for such commonly used distributions as the normal and its relatives, while special methods have been developed for some less common distributions; approximation methods are also used.

Quality and portability are important considerations when selecting a pseudo-random number generator. Some generators have been judged much better than others at simulating the desired distribution in a given computer/compiler environment. Some generators produce acceptable results in each of several environments, while others produce exactly the same results in different environments. While the latter may be desirable when simulation studies are to be performed in several environments, a price is paid in execution time.

The software cataloged here provides capabilities for a wide variety of univariate distributions (classified in alphabetical order), and several multivariate distributions.

January 1984

References

1. Kennedy, W.J., and J.E. Gentle (1980). *Statistical Computing*, Marcel Dekker, New York.

L6 : Pseudo-random number generation

L6a : Univariate

L6a2 : Beta, binomial, Boolean

DATAPAC subprogram library

BETRAN Generates a random sample of size N from the beta distribution with parameters ALPHA and BETA.

BINRAN Generates a random sample of size N from the binomial distribution with parameters P and NPAR.

IMSL subprogram library

GGBN Binomial random deviate generator.

GGBTR Beta random deviate generator.

MINITAB interactive system

BRANDOM Generates K pseudo-random numbers from binomial distribution (number of successes in n Bernoulli trials with probability p of success).

BTRIALS Generates pseudo-random sequence of K 0's and 1's, with the probability p of a 1.

NAG subprogram library

G05DLE Pseudo-random real numbers, Beta distribution of the first kind. Double precision version is G05DLF.

- G05DME** Pseudo-random real numbers, Beta distribution of the second kind. Double precision version is G05DMF.
- G05DZE** Pseudo-random logical value. Double precision version is G05DZF.
- G05EDE** Set up reference vector for generating pseudo-random integers, binomial distribution. Double precision version is G05EDF.

L6a3 : Cauchy, chi-squared

DATA PAC subprogram library

- CAURAN** Generates a random sample of size N from the Cauchy distribution with median = 0 and 75% point = 1.
- CHSRAN** Generates a random sample of size N from the chi-squared distribution with integer degrees of freedom parameter = NU.

IMSL subprogram library

- GGCAY** Cauchy random deviate generator.
- GGCHS** Chi-squared random deviate generator.

NAG subprogram library

- G05DFE** Pseudo-random real numbers, Cauchy distribution. Double precision version is G05DFF.
- G05DHE** Pseudo-random real numbers, chi-square distribution. Double precision version is G05DHF.

L6a4 : Discrete, double exponential

DATA PAC subprogram library

- DEXRAN** Generates a random sample of size N from the double exponential (Laplace) distribution with mean = 0 and standard deviation = $\sqrt{2}$.

MINITAB interactive system

- DRANDOM** Generates K pseudo-random numbers from a user-specified discrete distribution.

L6a5 : Exponential, extreme value

DATA PAC subprogram library

- EV1RAN** Generates a random sample of size N from the extreme value type 1 distribution with mean = Euler's number = 0.57721566.
- EV2RAN** Generates a random sample of size N from the extreme value type 2 distribution with tail length parameter = GAMMA.
- EXPRAN** Generates a random sample of size N from the exponential distribution with mean = 1 and standard deviation = 1.

IMSL subprogram library

- GGEXN** Exponential random deviate generator.
- GGEXT** Random deviate generator for a mixture of two exponentials.

NAG subprogram library

G05DBE Pseudo-random real numbers, exponential distribution. Double precision version is G05DBF.

L6a6 : F distribution

DATAPAC subprogram library

FRAN Generates a random sample of size N from the F-distribution with degrees of freedom parameters = NU1 and NU2.

NAG subprogram library

G05DKE Pseudo-random real numbers, Snedecor's F-distribution. Double precision version is G05DKF.

L6a7 : Gamma, general (continuous, discrete) distributions, geometric

DATAPAC subprogram library

GAMRAN Generates a random sample of size N from the gamma distribution with tail length parameter = GAMMA, mean = GAMMA and standard deviation = $\sqrt{\text{GAMMA}}$.

GEORAN Generates a random sample of size N from the geometric distribution with parameter P.

IMSL subprogram library

GGAMR One parameter gamma random deviate generator, and usable as basis for 2 parameter gamma, exponential, chi-squared, chi, beta, t and F deviate generator.

GGDA General discrete distribution random deviate generator using alias method.

GGDT General discrete distribution random deviate generator using table lookup method.

GGEOT Geometric random deviate generator.

GGVCR General continuous distribution random deviate generator.

NAG subprogram library

G05DGE Pseudo-random real numbers, gamma distribution with parameters (g,h). Double precision version is G05DGF.

G05EXE Set up reference vector from supplied cumulative distribution function or probability distribution function. Double precision version is G05EXF.

L6a8 : Halfnormal, hypergeometric

DATAPAC subprogram library

HFNRAN Generates a random sample of size N from the halfnormal distribution with mean = $\sqrt{2/\pi}$ and standard deviation = 1.

IMSL subprogram library

GGHPR Hypergeometric random deviate generator.

NAG subprogram library

G05EFE Set up reference vector for generating pseudo-random integers, hypergeometric distribution. Double precision version is G05EFF.

L6a9 : Integers**MINITAB interactive system****IRANDOM** Generates K pseudo-random integers in a specified interval.**NAG subprogram library****G05DYE** Pseudo-random integer from uniform distribution. Double precision version is G05DYF.**G05EBE** Set up reference vector for generating pseudo-random integers, uniform distribution. Double precision version is G05EBF.**G05EYE** Pseudo-random integer from reference vector. Double precision version is G05EYF.**L6a12 :** Lambda, logical, logistic, lognormal**DATAPAC subprogram library****LAMRAN** Generates a random sample of size N from the (Tukey) lambda distribution with tail length parameter ALAMBA.**LGNRAN** Generates a random sample of size N from the lognormal distribution with mean = sqrt(e).**LOGRAN** Generates a random sample of size N from the logistic distribution with mean = 0 and standard deviation = pi/sqrt(3).**IMSL subprogram library****GGNLTG** Log-normal random deviate generator.**NAG subprogram library****G05DCE** Pseudo-random real numbers, logistic distribution. Double precision version is G05DCF.**G05DEE** Pseudo-random real numbers, lognormal distribution. Double precision version is G05DEF.**G05DZE** Pseudo-random logical value. Double precision version is G05DZF.**L6a14 :** Negative binomial, normal**CMLIB subprogram library (FNLIB sublibrary)****RGAUSS** Normal random number.**CMLIB subprogram library (RV sublibrary)****RNOR** Generates quasi normal random numbers with zero mean and unit standard deviation.**DATAPAC subprogram library****NBRAN** Generates a random sample of size N from the negative binomial distribution with parameters P and NPAR.**NORRAN** Generates a random sample of size N from the normal (Gaussian) distribution with mean = 0 and standard deviation = 1.**IMSL subprogram library****GGBNR** Negative binomial random deviate generator.**GGNML** Normal or Gaussian random deviate generator.**GGNO** Generate set of order statistics from normal distribution.

GGNPM Normal random deviate generator via the polar method.
GGNQF Normal random deviate generator. Function form of GGNML.

MINITAB interactive system

NRANDOM Generates K pseudo-random numbers from the normal distribution with specified mean and standard deviation.

NAG subprogram library

G05DDE Pseudo-random real numbers, normal distribution (a,b). Double precision version is G05DDF.
G05EEE Set up reference vector for generating pseudo-random integers, negative binomial distribution. Double precision version is G05EEF.

STATLIB subprogram library

NRAND Generates a vector of normally distributed pseudo-random numbers.

L6a15 : Order statistics

IMSL subprogram library

GGNO Generate set of order statistics from normal distribution.
GGUO Generate set of order statistics from uniform (0,1) distribution.

L6a16 : Pareto, permutations, Poisson

DATAPAC subprogram library

PARRAN Generates a random sample of size N from the Pareto distribution with tail length parameter GAMMA.
POIRAN Generates a random sample of size N from the Poisson distribution with mean = ALAMBA and standard deviation = sqrt(ALAMBA).
RANPER Generates a random permutation of size N of the values 1.0, 2.0, 3.0, ..., N-1, N.

IMSL subprogram library

GGNPP Nonhomogeneous Poisson process generator with rate function lambda(t) - fixed interval, fixed number, or one at a time.
GGPER Generate a random permutation of the integers 1 to k.
GGPON Poisson random deviate generator where the Poisson parameter changes frequently.
GGPOS Poisson random deviate generator where the Poisson parameter does not change often.

MINITAB interactive system

PRANDOM Generates K pseudo-random numbers from the Poisson distribution with specified population mean K.

NAG subprogram library

G05ECE Set up reference vector for generating pseudo-random integers, Poisson distribution. Double precision version is G05ECF.
G05EHE Performs a pseudo-random permutation of a vector of integers. Double precision version is G05EHF.

L6a19 : Samples, stable distribution

IMSL subprogram library

GGSRs Generate a simple random sample from a finite population.

GGSTA Stable distribution random deviate generator.

MINITAB interactive system

SAMPLE Randomly selects without replacement values from one or more vectors, optionally carrying along other vectors.

NAG subprogram library

G05EJE Selects a pseudo-random sample from an integer vector. Double precision version is G05EJF.

L6a20 : t distribution, time series, triangular

DATAPAC subprogram library

TRAN Generates a random sample of size N from the Student's t distribution with degrees of freedom parameter NU.

IMSL subprogram library

FTGEN Generation of a time series from a given ARIMA (Box-Jenkins) stochastic model.

GGTRA Triangular distribution random deviate generator.

NAG subprogram library

G05DJE Pseudo-random real numbers, Student's t distribution. Double precision version is G05DJF.

G05EGE Set up reference vector for univariate ARMA time series model. Double precision version is G05EGF.

G05EWE Generate next term from ARMA time series using vector from G05EGE. Double precision version is G05EWF.

L6a21 : Uniform

CMLIB subprogram library (FNLIB sublibrary)

RAND Uniform random number on [0,1].

RUNIF Sequence of uniform random numbers on [0,1].

CMLIB subprogram library (RV sublibrary)

UNI Generates uniformly distributed random numbers on the interval [0,1]. UNI's main advantages are a long cycle and a high degree of reproducibility on other machines (it runs on any machine with at least 16 bit integer arithmetic).

DATAPAC subprogram library

UNIRAN Generates a random sample of size N from the uniform (rectangular) distribution on the unit interval (0,1).

IMSL subprogram library

GGUBFS Basic uniform (0,1) random number generator. Function form of GGUBS.

GGUBS Basic uniform (0,1) pseudo-random number generator.

GGUBT Uniform (0,1) pseudo-random number generator using alternate multiplier.

GGUD Discrete uniform random number generator.
GGUO Generate set of order statistics from uniform (0,1) distribution.
GGUW Uniform (0,1) random number generator with shuffling.

MINITAB interactive system

URANDOM Generates K pseudo-random numbers from the uniform (0,1) distribution.

NAG subprogram library

G05CAE Pseudo-random real numbers, uniform distribution over (0.0,1.0). Double precision version is G05CAF.
G05DAE Pseudo-random real numbers, uniform distribution over (a,b). Double precision version is G05DAF.
G05DYE Pseudo-random integer from uniform distribution. Double precision version is G05DYF.
G05EBE Set up reference vector for generating pseudo-random integers, uniform distribution. Double precision version is G05EBF.

PORT subprogram library

RANBYT Returns the real random variate generated by UNI, together with its bit pattern presented in four 8-bit bytes.
UNI Returns a single real random variate from the uniform [0,1) distribution.

L6a22 : Von Mises

IMSL subprogram library

GGVMS Von Mises random deviate generator.

L6a23 : Weibull

DATAPAC subprogram library

WEIRAN Generates a random sample of size N from the Weibull distribution with tail length parameter GAMMA.

IMSL subprogram library

GGWIB Weibull random deviate generator.

NAG subprogram library

G05DPE Pseudo-random real numbers, Weibull distribution. Double precision version is G05DPF.

L6b : Multivariate

IMSL subprogram library

ZSRCH Generate points in an n dimensional space.

L6b3 : Contingency table, correlation matrix

IMSL subprogram library

- GGCOR** Generate a random orthogonal matrix and a random correlation matrix.
- GGTAB** Generate a random contingency table with given row and column totals.

L6b13 : Multinomial**IMSL subprogram library**

- GGMTN** Multinomial random deviate generator.

L6b14 : Normal**IMSL subprogram library**

- GGNSM** Multivariate normal random deviate generator with given covariance matrix.

NAG subprogram library

- G05EZE** Returns a pseudo-random multivariate normal vector taken from a distribution described by a reference vector set up by G05EAE. Double precision version is G05EZF.

L6b15 : Orthogonal matrix**IMSL subprogram library**

- GGCOR** Generate a random orthogonal matrix and a random correlation matrix.

L6b21 : Uniform**IMSL subprogram library**

- GGSPH** Generation of uniform random deviates from the surface of the unit sphere in 3 or 4 space.

L6c : Service routines (e.g., seed)**NAG subprogram library**

- G05CBE** Initialise random number generating routines, to give a repeatable sequence. Double precision version is G05CBF.
- G05CCE** Initialise random number generating routines, to give nonrepeatable sequence. Double precision version is G05CCF.
- G05CFE** Save state of random number generating routines. Double precision version is G05CFF.
- G05CGE** Restore state of random number generating routines. Double precision version is G05CGF.

PORT subprogram library

- RANSET** Initializes the uniform random number generator, UNI, to other than the default initial values.

Computer implementation of analyses of experimental designs is straightforward, and the output is commonly clear, when the number of observations is constant across all treatment level combinations. Care should be exercised when the number of observations varies across cells; some output values may be computed differently than expected, and, for the analysis, ordinary interpretation of classical tests may not apply.

Most statistical analyses of experimental designs assume independent, normally distributed errors with zero mean and constant variance. The appropriateness of assumptions such as these should be evaluated through statistical and graphical analysis. Transformation of the data may make assumptions more reasonable. Less restrictive assumptions are needed for nonparametric (distribution-free) analyses.

Software in this chapter is first distinguished by data type - univariate and multivariate (though there are currently no multivariate codes classified). The next level distinguishes among forms of experimental designs. Software is classified both at the leaves of the tree and at higher levels in this chapter; the latter performs more than one of the tasks in the subchapters.

MATLAB, an interactive system, has analysis of variance capabilities (see the Library Reference).

January 1984

References

1. Box, G.E.P., W.G. Hunter, and J.S. Hunter (1978). *Statistics for Experimenters*, John Wiley & Sons, New York.
2. Cochran, W.G., and G.M. Cox (1957). *Experimental Design*, John Wiley & Sons, New York.
3. Cox, D.R. (1958). *Planning of Experiments*, John Wiley & Sons, New York.
4. John, P.W.M. (1971). *Statistical Design and Analysis of Experiments*, Macmillan, New York.
5. Natrella, M.G. (1963). *Experimental Statistics*, NBS Handbook 91, U.S. Government Printing Office, Washington, D.C.
6. Searle, S.R. (1971). *Linear Models*, John Wiley & Sons, New York.

L7 : Experimental design, including analysis of variance

BMDP program library

P4V Interactive or batch univariate and multivariate ANOVA and ANCOVA, including nested, repeated measures, split-plot, and changeover designs, and model building features. Options: cell weights for hypothesis testing, contrasts, tests of simple effects, save cell means.

L7a : Univariate

BMDP program library

P2V ANOVA and ANCOVA for unbalanced fixed-effect models (including full and fractional factorial designs, Latin squares), and repeated measure models, or a combination of models, with Greenhouse-Geisser and Huynh-Feldt deg. of freedom adjustment. Options: orthogonal decomposition of within-effects, save results.

L7a1 : One-way analysis of variance

BMDP program library

P7D Side-by-side histograms for each cell in one-way or two-way ANOVA, within-group summary statistics and ANOVA table (with equality of variance test and tests that do not assume equal variances). Options: trimmed mean analysis, ANOVA diagnostics, tests of pairwise mean comparisons, correlations, Winsorized means.

P9D Provides descriptive statistics (means, std. devs., frequencies, one-way ANOVA table) of groups (cells) for data classified into cells using one or more grouping variables. Options: miniplots of cell means (eight per page), plot frames are defined by combinations of levels of grouping variables.

L7a1a : Parametric analysis**BMDP program library**

P1V Performs one-way ANOVA or ANCOVA with standard results. For ANCOVA, tests 1) equality of slopes, 2) zero slope, and 3) equality of adjusted cell means; plots the covariate for each group. Tests equality of pairs of means (or adjusted means). Options: linear contrasts, within-group correlations and statistics.

IMSL subprogram library

ACRDAN Analysis of one-way classification design data.

MINITAB interactive system

AOVONEWAY Performs a one-way analysis of variance and prints standard results.

ONEWAY Performs one-way analysis of variance, prints standard results, and optionally saves results.

NAG subprogram library

G04AEE One-way analysis of variance, subgroups of unequal size. Double precision version is G04AEF.

STATLIB subprogram library

ONEWY Performs one-way analysis of variance for two or more random samples.

ONEWYS Performs one-way analysis of variance for two or more random samples, with computed results returned to the user.

L7a1a1 : Contrasts, multiple comparisons**IMSL subprogram library**

ACTRST Contrast estimates and sums of squares.

ASNKMC Student-Newman-Keuls multiple comparison test.

L7a1a2 : Analysis of variance components**IMSL subprogram library**

AGVACL One or two-sided interval estimate of a variance component.

L7a1b : Distribution-free (nonparametric) analysis**IMSL subprogram library**

NAWRPE Wilsons ANOVA (1, 2, or 3 way designs) with equal replication.

NAWRPU Wilsons ANOVA (1, 2, or 3 way designs) with unequal replication.

NAG subprogram library

G08AFE Kruskal-Wallis 1-way analysis of variance on k samples of unequal size. Double precision version is G08AFF.

L7a2 : Balanced multiway design

L7a2a : Complete**BMDP program library**

- P8V** ANOVA for complete designs with equal cell sizes – nested, crossed, partially nested, partially crossed designs for fixed-effect models, mixed models (including repeated measures), and random-effect models, with parameter estimation and printing.
- P9D** Provides descriptive statistics (means, std. devs., frequencies, one-way ANOVA table) of groups (cells) for data classified into cells using one or more grouping variables. Options: miniplots of cell means (eight per page), plot frames are defined by combinations of levels of grouping variables.

L7a2a1 : Parametric analysis**IMSL subprogram library**

- AGBACP** Analysis of balanced complete experimental design structure data.
- AGXPM** Expected mean squares for balanced complete design models.
- AORDR** Reordering of the data obtained from any balanced complete experimental design.

L7a2a1a : Two-way**BMDP program library**

- P7D** Side-by-side histograms for each cell in one-way or two-way ANOVA, within-group summary statistics and ANOVA table (with equality of variance test and tests that do not assume equal variances). Options: trimmed mean analysis, ANOVA diagnostics, tests of pairwise mean comparisons, correlations, Winsorized means.

IMSL subprogram library

- ARCBAN** Analysis of two-way classification design data.

MINITAB interactive system

- TWOWAYAOV** Performs two-way analysis of variance for balanced data (equal number of observations, one or more, in each cell) and prints standard results. Options: fit additive model, save results.

NAG subprogram library

- G04AFE** Two-way analysis of variance, cross-classification, subgroups of equal size. Double precision version is G04AFF.

L7a2a1b : Factorial**IMSL subprogram library**

- AFACN** Full factorial plan analysis.
- **AFACT** Full factorial plan analysis - easy to use version.

L7a2a1c : Nested**IMSL subprogram library**

- ANESTE** Analysis of completely nested design data with equal numbers in the subclasses.
ANESTU Analysis of completely nested design data with unequal number in the subclasses.

L7a2a2 : Distribution-free (nonparametric) analysis

BMDP program library

- P3S** Computes and prints results from one or more of the following: sign test, Wilcoxon signed-rank test, Mann-Whitney rank-sum test, Kruskal-Wallis one-way ANOVA, Friedman two-way ANOVA, Kendall's coefficient of concordance, Kendall and Spearman rank-correlation coefficients.

IMSL subprogram library

- NAFRE** Friedman's test for randomized complete block designs.
NAWNRPE Wilcoxon ANOVA (2 or 3 way designs) without replicates.
NAWRPE Wilcoxon ANOVA (1, 2, or 3 way designs) with equal replication.

NAG subprogram library

- G08AEE** Friedman 2-way analysis of variance on k matched samples. Double precision version is G08AEF.

L7a2b : Incomplete

L7a2b1 : Parametric analysis

IMSL subprogram library

- ABIBN** Analysis of balanced incomplete block and balanced lattice designs.

L7a2b1a : Latin square

IMSL subprogram library

- ALSQAN** Analysis of Latin square design data.

NAG subprogram library

- G04ADE** Three-way analysis of variance, Latin square design. Double precision version is G04ADF.

L7a2b1b : Lattice designs

L7a2b2 : Distribution-free (nonparametric) analysis

L7a3 : Analysis of covariance

BMDP program library

- P1V** Performs one-way ANOVA or ANCOVA with standard results. For ANCOVA, tests 1) equality of slopes,

2) zero slope, and 3) equality of adjusted cell means; plots the covariate for each group. Tests equality of pairs of means (or adjusted means). Options: linear contrasts, within-group correlations and statistics.

IMSL subprogram library

ANCOV1 Covariance analysis for one-way classification design data.

L7a4 : General linear model (unbalanced design)

BMDP program library

P3V Uses maximum likelihood (ML) and restricted ML approaches to balanced and unbalanced fixed and random coefficient models of quite arbitrary form (including having covariates), with parameter estimation, hypothesis testing, and printing. Weights optional.

L7a4a : Parametric analysis

IMSL subprogram library

AGLMOD General linear model analysis.

AMEANS Preparation of a set of unbalanced data for analysis by the method of unweighted means.

NAG subprogram library

G04AGE Two-way analysis of variance, hierarchical classification, subgroups of unequal size. Double precision version is G04AGF.

L7a4b : Distribution-free (nonparametric) analysis

IMSL subprogram library

NAWRPU Wilsons ANOVA (1, 2, or 3 way designs) with unequal replication.

L7b : Multivariate

L8: Regression

Regression analysis is concerned with estimating the relationship between one "dependent" variable and one or more "independent" variables. The quality of the estimate is commonly judged by evaluating a function of the residuals (the differences between the observations of the dependent variable and their estimates). Least squares regression provides the best linear unbiased estimates of the regression parameters when the errors are independent random variables with zero mean and constant variance. Techniques such as L_1 , L_∞ , robust, or EDA (exploratory data analysis) regression may be more appropriate when these distributional assumptions are not valid; other alternatives are appropriate when the independent variables are also subject to error, when the errors are autocorrelated, and so on.

Numerical problems arise in multiple regression when the design matrix is ill-conditioned; orthogonal polynomials provide one solution to this problem in the case of polynomial regression, while elimination of independent variables and ridge regression provide possible solutions in the general case. Variable selection techniques are useful in selecting one or more "best" regression equations given a number of candidate independent variables.

Regression software comes in the form of subroutines (IMSL and NAG have both building blocks and full regression subroutines. STATLIB subroutines provide comprehensive printing of output, CMLIB packages provides several special-purpose regression capabilities), programs (nine in BMDP, two in INVAR), and interactive system commands (four in

Minitab).

The software in this chapter is first organized by the type of regression (e.g., linear least squares regression (L8a), EDA regression (L8f), nonlinear regression (L8g)). The bulk of the software is in chapter L8a, and is distinguished either by the nature of the independent variable data or by the purpose (e.g., simple (L8a1), multiple (L8a4), regression diagnostics (L8a9)). In addition to doing regression, much of the software in these sections also provides hypothesis testing, inference, and analysis of residuals, so that chapters L8a9 and L8a10 (for hypothesis testing and inference) contain software that provides only diagnostics or testing and inference.

Future editions of GAMS will include software for robust regression and for total least squares (errors in variables) regression, and addition nonlinear regression codes.

January 1984

References

1. Daniel, C., and F.S. Wood (1980). *Fitting Equations to Data*, Wiley-Interscience, New York.
2. Draper, N. R. and H. Smith (1981). *Applied Regression Analysis*, Wiley Interscience, New York.
3. Kelly, L.G. (1967). *Handbook of Numerical Methods and Applications*, Addison-Wesley, Reading, Massachusetts.
4. Mosteller, F., and J.W. Tukey (1977). *Data Analysis and Regression*, Addison-Wesley, Reading, Massachusetts.
5. Neter, J., and W. Wasserman (1974). *Applied Linear Statistical Models*, Richard D. Irwin, Inc., Homewood, Illinois.
6. Seber, G.A.F. (1977). *Linear Regression Analysis*, John Wiley & Sons, New York.

L8 : Regression (*search also classes G, K*)

L8a : Linear least squares (L-2) (*search also classes D5, D6, D9*)

MINITAB interactive system

REGRESS Performs simple or multiple linear regression, prints standard results. Options: amount of output, save results, weights, handle missing values, through the origin, compute and save regression diagnostics, lack of fit tests.

L8a1 : Simple

L8a1a : Ordinary

L8a1a1 : Unweighted

L8a1a1a : No missing values

IMSL subprogram library

RLONE Analysis of a simple linear regression model.

NAG subprogram library

G02CAE Simple linear regression with constant term, no missing values. Double precision version is G02CAF.

L8a1a1b : Missing values

IMSL subprogram library

- BEMIRI** Estimates means, simple regression coefficients, their intercepts, standard errors of the regression coefficients, and standard deviations for arrays which contain missing values (in-core version).
- BEMIRO** Estimates means, simple regression coefficients, their intercepts, standard errors of the regression coefficients, and standard deviations for arrays which contain missing values (out-of-core version).

NAG subprogram library

- G02CCE** Simple linear regression with constant term, missing values. Double precision version is G02CCF.

L8a1a2 : Weighted

L8a1b : Through the origin

NAG subprogram library

- G02CBE** Simple linear regression without constant term, no missing values. Double precision version is G02CBF.
- G02CDE** Simple linear regression without constant term, missing values. Double precision version is G02CDF.

L8a1c : Errors in variables

CMLIB subprogram library (SLRPACK sublibrary)

- RGM** Computes estimates of simple linear regression parameters for a geometric mean regression.
- RYORK** Estimates simple linear regression coefficients when both variables are subject to errors which are not necessarily homogeneous in variance.

L8a1d : Calibration (inverse regression)

IMSL subprogram library

- RLINCF** Response control using a fitted simple linear regression model.

L8a2 : Polynomial

L8a2a : Not using orthogonal polynomials

DATAPAC subprogram library

- POLY** Computes a least squares polynomial fit (of degree = IDEG) of the response variable data in the vector Y as a function of vector X and with optional weights.

L8a2a1 : Unweighted

L8a2a2 : Weighted

L8a2b : Using orthogonal polynomials**BMDP program library**

P5R Least squares fit of a polynomial in one independent variable to the dependent variable. Prints standard results and goodness-of-fit statistics for each polynomial degree. Computations use orthogonal polynomials. Options: weights, additional printing, and three plots.

IMSL subprogram library

• **RLFOR** Fit a univariate curvilinear regression model using orthogonal polynomials with optional weighting and prediction analysis - easy-to-use version.

RLPOL Generate orthogonal polynomials with the associated constants AA and BB.

L8a2b1 : Unweighted**IMSL subprogram library**

RLDCVA Variance estimates for decoded orthogonal polynomial regression coefficients.

RLFOTH Fit a univariate curvilinear regression model using orthogonal polynomials.

STATLIB subprogram library

PFIT Performs linear least squares regression analysis of a polynomial model.

PFITS Performs linear least squares regression analysis of a polynomial model, with computed results returned to the user.

L8a2b2 : Weighted**IMSL subprogram library**

RLDCVA Variance estimates for decoded orthogonal polynomial regression coefficients.

RLFOTW Fit a univariate curvilinear regression model using orthogonal polynomials with weighting.

STATLIB subprogram library

PFITW Performs weighted linear least squares regression analysis of a polynomial model.

PFITWS Performs weighted linear least squares regression analysis of a polynomial model with computed results returned to the user.

L8a3 : Piecewise polynomial (i.e. multiphase or spline)**CMLIB subprogram library (FC sublibrary)**

FC Fits piecewise polynomial to discrete data with equality and inequality constraints.

L8a4 : Multiple**BMDP program library**

P1R Performs multiple linear regression and prints standard results. Options: weights, form of input, regression on subsets or groups and test of equality of regression lines, intercept term present or absent,

more printing, five plots, save predicted values and residuals.

P2R

Multiple linear regression, with standard results. Options: weights, forward or backward stepping, interactive stepping, stepping sets of variables (e.g. design variables), forcing variables into the model, eleven diagnostics (including Cook and AP statistics) available for printing, plotting, and saving.

L8a4a : Ordinary

L8a4a1 : Unweighted

L8a4a1a : No missing values

STATLIB subprogram library

FIT

Performs linear least squares regression analysis of a general linear model.

FITS

Performs linear least squares regression analysis of a general linear model with computed results returned to the user.

L8a4a1b : Missing values

L8a4a1c : From correlation data

IMSL subprogram library

RLMUL

Multiple linear regression analysis.

NAG subprogram library

G02CGE

Performs a multiple linear regression on the set of variables whose means, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients are given. Double precision version is G02CGF.

G02CHE

Performs a multiple linear regression with no constant on the set of variables whose sums of squares and cross-products about zero and correlation-like coefficients are given. Double precision version is G02CHF.

L8a4a1d : Using principal components

BMDP program library

P4R

Regression analysis for a dependent variable on a set of principal components computed from the independent variables in a stepwise manner determined either by magnitude of eigenvalue or correlations between dependent variable and components, with printing. Options: form of input, more printing, four plots.

L8a4a1e : Using preference pairs

BMDP program library

P9M

Scoring based on preference pairs - for each observation construct score as linear combination of

variables with coefficients based on expert preference, in stepwise manner. Options: printing, plots, compare results when analysis is repeated for different judges.

L8a4a2 : Weighted

STATLIB subprogram library

FITW Performs weighted linear least squares regression analysis of a general linear model.

FITWS Performs weighted linear least squares regression analysis of a general linear model with computed results returned to the user.

L8a4b : Errors in variables

L8a4d : Logistic

BMDP program library

PLR Stepwise logistic regression for binary dependent variable and categorical (design variables are formed) and continuous independent variables, using either maximum likelihood or approximate asymptotic estimates for stepping. Three options for generating design variables, plots, SYSTEM WARNING - MAX PAGES interactive stepping.

L8a5 : Variable selection

BMDP program library

P2R Multiple linear regression, with standard results. Options: weights, forward or backward stepping, interactive stepping, stepping sets of variables (e.g. design variables), forcing variables into the model, eleven diagnostics (including Cook and AP statistics) available for printing, plotting, and saving.

P9R Estimates regression equations for "best" (by R-squared, adjusted R-squared, or Mallows' C(p) criterion) subset of predictor variables by Furnival-Wilson algorithm. Options: weights, form of input, Durbin-Watson statistic. Cook's distance and several types of residuals may be printed, plotted, or saved.

IMSL subprogram library

RLEAP Leaps and bounds algorithm for determining a number of best regression subsets from a full regression model. USLEAP is a special purpose output routine designed to be used only in conjunction with RLEAP.

RLSEP Selection of a regression model using a forward stepwise algorithm, and computation of the usual analysis of variance table entries - easy-to-use version.

RLSTP Regression model selection using a forward stepwise algorithm with results available after each step.

MINITAB interactive system

STEPWISE Performs stepwise linear regression using forward selection, backward elimination, conventional stepwise, or user intervention. Options available through subcommands: F-to-enter and F-to-remove, force and remove sets of variables, print next "best" (by the F-statistic) K alternatives.

I 8a6 : Regression design**IMSL subprogram library****RLCOMP** Generation of an orthogonal central composite design.**L8a7 :** Several multiple regressions**BMDP program library****P1R** Performs multiple linear regression and prints standard results. Options: weights, form of input, regression on subsets or groups and test of equality of regression lines, intercept term present or absent, more printing, five plots, save predicted values and residuals.**P6R** Computes the partial correlations of a set of variables after removing the linear effects of a second set of variables. Can be used for regression, especially if multiple dependent variables are present. Prints standard results. Options: weights, form of input, additional printing and plots.**NAG subprogram library****G02CJE** Performs one or more multiple linear regressions, regressing each of a set of dependent variables separately on the same set of independent variables. Input is raw data; output includes, for each dependent variable, estimates of regression coefficients and an estimate of the variance of residuals. Double precision version is G02CJF.**L8a8 :** Multivariate**IMSL subprogram library****OFIMA3** Least squares solution to the matrix equation $AT = B$.**L8a9 :** Diagnostics**BMDP program library****P2R** Multiple linear regression, with standard results. Options: weights, forward or backward stepping, interactive stepping, stepping sets of variables (e.g. design variables), forcing variables into the model, eleven diagnostics (including Cook and AP statistics) available for printing, plotting, and saving.**P9R** Estimates regression equations for "best" (by R-squared, adjusted R-squared, or Mallows' C(p) criterion) subset of predictor variables by Furnival-Wilson algorithm. Options: weights, form of input, Durbin-Watson statistic. Cook's distance and several types of residuals may be printed, plotted, or saved.**MINITAB interactive system****REGRESS** Performs simple or multiple linear regression, prints standard results. Options: amount of output, save results, weights, handle missing values, through the origin, compute and save regression diagnostics, lack of fit tests.**L8a10 :** Hypothesis testing, inference

L8a10a : Lack-of-fit tests

IMSL subprogram library

- RLFITI** Pure replication error degrees of freedom and sum of squares - in-core version.
RLFITO Pure replication error degrees of freedom and sum of squares - out-of-core version.

L8a10b : Analysis of residuals

IMSL subprogram library

- RLRES** Perform a residual analysis for a fitted regression model.

L8a10c : Inference

IMSL subprogram library

- RLINPF** Inverse prediction using a fitted simple linear regression model.
RLOPDC Response prediction using an orthogonal polynomial regression model.
RLPRDI Confidence intervals for the true response and for the average of a set of future observations on the response - in-core version.
RLPRDO Confidence intervals for the true response and for the average of a set of future observations on the response - out-of-core version.

L8b : Biased (ridge)

BMDP program library

- P3R** Performs nonlinear least squares regression with standard results. Six functions are built in; others can be specified. Options: weights, evaluates functions of parameters (with standard errors), upper and lower limits on parameters, ridging, exact linear constraints, maximum likelihood estimates, and five plots.
PAR Performs nonlinear regression using pseudo-Gauss-Newton algorithm. Derivatives are NOT specified. Options: weights, linear inequality constraints, maximum likelihood, functions of parameters, ridging, four plots, fitting models defined by differential equations.

L8c : Linear least absolute value (L-1)

IMSL subprogram library

- RLLMV** Perform linear regression using the minimax criterion.

L8d : Linear minimax (L-infinity)

IMSL subprogram library

- RLLAV** Perform linear regression using the least absolute values criterion.

L8e : Robust

L8f : EDA

MINITAB interactive system

RLINE Fits straight line to x-y data by resistant line procedure – partitions data by x-value into three groups and uses an iterative procedure to find the line that makes the median residual in the left and the right partitions equal.

RSMOOTH Computes resistant smoother by 4253H, twice (or 3RSSH, twice), i.e. successive application of running medians and Hanning (running weighted averages), and save results.

L8g : Nonlinear

INVAR program library

INVAR1 Interactive program for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results. Line printer graphics only.

INVAR2 Interactive program for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results and DISPLA graphics.

L8g1 : Unweighted

L8g1a : Derivatives not supplied

BMDP program library

PAR Performs nonlinear regression using pseudo-Gauss-Newton algorithm. Derivatives are NOT specified. Options: weights, linear inequality constraints, maximum likelihood, functions of parameters, ridging, four plots, fitting models defined by differential equations.

CMLIB subprogram library (NL2SN sublibrary)

NL2SN Minimizes a nonlinear sum of squares using residual values only. Double precision version is DNL2SN.

IMSL subprogram library

RSMITZ Least squares fit of the non-linear regression model $y(i) = \alpha + \beta \cdot \gamma^{x(i)} + e(i)$.

ZXSSQ Minimum of the sum of squares of m functions in n variables using a finite difference Levenberg-Marquardt algorithm.

STATLIB subprogram library

NFIT Performs nonlinear least squares regression analysis using the Levenberg-Marquardt algorithm.

NFITS Performs nonlinear least squares regression analysis using the Levenberg-Marquardt algorithm, with results returned to the user.

L8g1b : Derivatives supplied

BMDP program library

P3R Performs nonlinear least squares regression with standard results. Six functions are built in; others can be specified. Options: weights, evaluates functions of parameters (with standard errors), upper and lower limits on parameters, ridging, exact linear constraints, maximum likelihood estimates, and five plots.

CMLIB subprogram library (NL2SN sublibrary)

NL2S1 Minimizes a nonlinear sum of squares using both residual and gradient values supplied by the user. Double precision version is DNL2S1.

L8g2 : Weighted

L8g2a : Derivatives not supplied

BMDP program library

PAR Performs nonlinear regression using pseudo-Gauss-Newton algorithm. Derivatives are NOT specified. Options: weights, linear inequality constraints, maximum likelihood, functions of parameters, ridging, four plots, fitting models defined by differential equations.

CMLIB subprogram library (NL2SN sublibrary)

NL2SN Minimizes a nonlinear sum of squares using residual values only. Double precision version is DNL2SN.

IMSL subprogram library

ZXSSQ Minimum of the sum of squares of m functions in n variables using a finite difference Levenberg-Marquardt algorithm.

STATLIB subprogram library

NFITW Performs weighted nonlinear least squares regression analysis using the Levenberg-Marquardt algorithm.

NFITWS Performs weighted nonlinear least squares regression analysis using the Levenberg-Marquardt algorithm, with results returned to the user.

L8g2b : Derivatives supplied

BMDP program library

P3R Performs nonlinear least squares regression with standard results. Six functions are built in; others can be specified. Options: weights, evaluates functions of parameters (with standard errors), upper and lower limits on parameters, ridging, exact linear constraints, maximum likelihood estimates, and five plots.

CMLIB subprogram library (NL2SN sublibrary)

NL2S1 Minimizes a nonlinear sum of squares using both residual and gradient values supplied by the user. Double precision version is DNL2S1.

L8h : Service routines

IMSL subprogram library

- RLDCQM** Decoding of a quadratic regression model.
- RLDCW** Variances of coded orthogonal polynomial regression coefficients - for usage in conjunction with IMSL routines RLFOTH and RLFOTW, and provided to prepare input for IMSL routine RLDCVA.
- RLDOPM** Coefficient decoder for an orthogonal polynomial regression model.
- RLSUBM** Retrieval of a symmetric submatrix from a matrix stored in symmetric storage mode by RLSTP.
- RLSUM** Reordering of the rows and corresponding columns of a symmetric matrix stored in symmetric storage mode.
- USLEAP** Print results of the best-regressions analysis performed by IMSL routine RLEAP.

NAG subprogram library

- G02CEE** Service routines for multiple linear regression, select elements from vectors and matrices. Double precision version is G02CEF.
- G02CFE** Service routines for multiple linear regression, re-order elements of vectors and matrices. Double precision version is G02CFF.

L9: Categorical Data Analysis

Categorical data analysis is concerned with the analysis of counts of observations assigned to categories (commonly displayed as contingency tables).

While special purpose software is available for analyzing two-by-two and two-way tables, software for the log-linear model allows analysis of higher dimensional tables, modeling, parameter estimation, analysis of residuals, and analysis of special types of tables (e.g., those with structural zeros). Some EDA techniques use the median instead of the mean as a measure of central tendency. Graphical displays of parameter estimates and of residuals are useful in analyzing the quality of a fitted model.

Software is classified both at the leaves of the tree and at higher levels in this chapter; the latter perform a variety of categorical data analysis computations. Software for summarizing data in tabular form is also classified in class L2b.

January 1984

References

1. Bishop, Y.M.M., S.E. Fienberg, and P.W. Holland (1975). *Discrete Multivariate Analysis*, MIT Press, Cambridge, Massachusetts.
2. Fienberg, S.E. (1977). *The Analysis of Cross-Classified Categorical Data*, MIT Press, Cambridge, Massachusetts.
3. Tukey, J.W. (1977). *Exploratory Data Analysis*, Addison-Wesley, Reading, Massachusetts.

L9 : Categorical data analysis

BMDP program library

- P4F** Forms two- or multi-way frequency tables. Options: percents; 25 tests and measures for two-way tables; fits and tests log-linear models, tests of marginal and partial association, stepwise models, three forms of input, structural zeros, cell and strata deletion, residuals.

MINITAB interactive system

- TABLE** Produces and prints one-way, two-way, and multi-way tables of counts with 20 optional subcommands for summarizing (e.g., cell mean, standard deviation), marginals, performing chi-square tests for each 2-way table, handling missing values, and selecting forms of input and output.

L9a : 2-by-2 tables**IMSL subprogram library****NHEXT** Fishers exact method for 2-by-2 tables.**L9b :** Two-way tables**IMSL subprogram library****BDTWT** Computations of a two-way frequency table.**CBNRHO** Estimation of the bivariate normal correlation coefficient using a contingency table.**CTPR** Compute exact probabilities for contingency tables.**CTRBYC** Analysis of a contingency table.**MINITAB interactive system****CHISQUARE** Performs chi-square test for association (non-independence) on a two-way table and prints standard results.**NAG subprogram library****G01AFE** Two-way contingency table analysis. Double precision version is G01AFF.**L9c :** Log-linear model**IMSL subprogram library****CTLLF** Log-linear fit of contingency table.**L9d :** EDA (e.g., median polish)**IMSL subprogram library****BEMDP** Median polish of a two-way table.**MINITAB interactive system****MPOLISH** Uses median polish to fit an additive model to a two-way layout which may be unbalanced and may have empty cells. Options: fit columns first, number of iterations, save results.**L10: Time Series Analysis**

Two widely used approaches to time series analysis are Box-Jenkins methods and spectral analysis. The Box-Jenkins approach seeks to build models with autoregressive terms (to model the "memory" of the underlying system) and moving average terms (to model the random effects). The method seems most appropriate when the source of the time series data is a structured system subject to random forcings. Business and economic time series have been successfully analyzed using this approach, which is currently also used in many engineering applications.

Spectral analysis, which had its origin in the work of Fourier, seeks to replace a given time series with its representation in terms of frequencies. Assuming that the time series can be modelled as a superposition of oscillations with the appropriate frequencies and amplitudes, spectral analysis seeks to isolate those characteristics. The method is most appropriate for series with strong deterministic components as well as noise. It is commonly used in physical science and engineering

applications, and also has been used successfully in biology and the social sciences.

The two methods are not mutually exclusive, and in fact share a number of techniques (e.g., autocorrelation analysis).

Subchapters in this chapter include time series data manipulation (e.g., filtering (L10b)), autocorrelation analysis (L10c), Box-Jenkins ARMA and ARIMA techniques (L10e), and spectral analysis (L10f). Cross-correlation analysis (L10g) deals with more than one time series.

The software in this chapter includes subroutines from CMLIB (MAXENTROPY sublibrary), DATAPAC, IMSL, NAG, and STATLIB, programs from BMDP, interactive system commands from Minitab, and the interactive system SPECAN (with high resolution graphics capabilities) from the Spectran library.

January 1984

References

1. Bloomfield, P. (1978). *Fourier Analysis of Time Series: An Introduction*, John Wiley & Sons, New York.
2. Box, G.E., and G.M. Jenkins (1978). *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco.
3. Fuller, W. (1970). *Introduction to Statistical Time Series*, John Wiley & Sons, New York.

L10 :	Time series analysis (<i>search also class L9c5 for time series graphics</i>)
--------------	---

L10a :	Transformations, transforms (<i>search also class J1</i>)
---------------	---

DATAPAC subprogram library

FOURIE Performs a Fourier analysis of the data in the input vector X.

IMSL subprogram library

FTRDIF Transformations, differences and seasonal differences of a time series for model identification.

MINITAB interactive system

DIFFERENCES Computes differences between observations at a specified lag in a time series.

LAG Computes lagged observations in a time series.

L10b :	Smoothing, filtering
---------------	----------------------

IMSL subprogram library

FTKALM Kalman filtering.

FTRDIF Transformations, differences and seasonal differences of a time series for model identification.

ICSMOU One-dimensional data smoothing by error detection.

NAG subprogram library

G13AAE Carries out non-seasonal and seasonal differencing on a time series. Information which allows the original series to be reconstituted from the differenced series is also produced. This information is required in time series forecasting. Double precision version is G13AAF.

G13BAE Filters a time series by an ARIMA model. Double precision version is G13BAF.

STATLIB subprogram library

MOVAVG Computes a k-term symmetric moving average of a series.

TSDIFF Performs a user-controlled differencing operation on a series.

L10c : Autocorrelation analysis**DATAPAC subprogram library****AUTO** Computes the sample autocorrelation coefficient of the data in the input vector X.**TIME** Performs a time series analysis on the data in the input vector X.**IMSL subprogram library****FTAUTO** Mean, variance, autocovariances, autocorrelations, and partial autocorrelations for a stationary time series.**MINITAB interactive system****ACF** Computes and graphs the autocorrelations of a time series, and optionally saves results.**PACF** Computes and graphs partial autocorrelations of a time series and optionally saves results.**NAG subprogram library****G13ABE** Computes the sample autocorrelation function of a time series. It also computes the sample mean, the sample variance and a statistic which may be used to test the hypothesis that the true autocorrelation function is zero. Double precision version is G13ABF.**G13ACE** Calculates partial autocorrelation coefficients given a set of autocorrelation coefficients. It also calculates the predictor error variance ratios for increasing order of finite lag autoregressive predictor, and the autoregressive parameters associated with the predictor of maximum order. Double precision version is G13ACF.**STATLIB subprogram library****ACORR** Performs autocorrelation analysis of a series.**ACORRD** Performs autocorrelation analysis of a series differenced with a user-controlled differencing operation.**ACORRS** Performs autocorrelation analysis of a series, with computed results returned to the user.**L10d :** Complex demodulation**DATAPAC subprogram library****DEM** Performs a complex demodulation on the data in the input vector X at the input demodulation frequency = F.**L10e :** ARMA and ARIMA modeling and forecasting**BMDP program library****P2T** Interactive or batch Box-Jenkins time series analysis for univariate time domain models (including ARIMA, regression, intervention, and transfer function models) – model identification, parameter estimation, testing, forecasting. Options: print, plot, differencing and filtering, save results.**IMSL subprogram library****FTCP** Non-seasonal ARIMA (Box-Jenkins) stochastic model analysis for a single time series with full parameter iteration and maximum likelihood estimation.**MINITAB interactive system****ARIMA** Fits non-seasonal and seasonal models to a time series with p the order of the AR part, d the number of differences, q the order of the MA part and with optional seasonality with period S, AR order P,

number of differences D, and MA order Q. Options: starting values, forecasting, save results.

L10e1 :	Model and parameter estimation
----------------	--------------------------------

IMSL subprogram library

- FTARPS** Preliminary estimation of the autoregressive parameters in an ARIMA stochastic model.
- FTMA** Preliminary estimation of the moving average parameters in an ARIMA stochastic model.
- FTML** Maximum likelihood estimation of autoregressive and moving average parameters in an ARIMA (Box-Jenkins) stochastic model.

NAG subprogram library

- G13ADE** Calculates preliminary estimates of the parameters of an autoregressive moving-average (ARMA) model from an autocorrelation function. Double precision version is G13ADF.
- G13AEE** Iteratively fits seasonal autoregressive-integrated moving-average (ARIMA) model to observed time series using non-linear least squares procedure incorporating backforecasting. Returns parameter estimates, standard errors, residual series, and information for use by G13AGE or G13AHE in forecasting. Double precision version is G13AEF.
- **G13AFE** Easy-to-use version of G13AEE. Iteratively fits seasonal ARIMA model to observed series using non-linear least squares procedure incorporating backforecasting. Returns parameter estimates, standard errors, residual series, and information for use by G13AGE or G13AHE in forecasting. Double precision version is G13AFF.

STATLIB subprogram library

- ARIMAE** Performs least squares estimation of the parameters in an ARIMA (Box-Jenkins) model using an adaptation of Pack's code.

L10e2 :	Forecasting
----------------	-------------

IMSL subprogram library

- FTCAST** Time series forecasts and probability limits using an ARIMA (Box-Jenkins) model.
- FTWEIN** Wiener forecast for a stationary stochastic process.

NAG subprogram library

- G13AGE** Accepts new observation of fully specified (by G13AEE or G13AFE) time series and updates "state set" (from G13AEE or G13AFE) information for use in further forecasting. Returns residuals corresponding to the new observations, which may be used in checking that new observations conform to previously fitted model. Double precision version is G13AGF.
- G13AHE** Produces forecasts of a time series, given model already fitted (by G13AEE or G13AFE). Original observations are not required, since the subroutine uses state set produced originally by G13AEE or G13AFE or updated by G13AGE. Standard errors of the forecasts are also provided. Double precision version is G13AHF.
- G13AJE** Applies a fully specified seasonal ARIMA model to an observed time series, generates the state set for forecasting and (optionally) derives a specified number of forecasts together with their standard deviations. Double precision version is G13AJF.

STATLIB subprogram library

- ARIMAF** Performs minimum mean square error forecasts for a given (fitted) ARIMA (Box-Jenkins) model, using an adaptation of Pack's code.

L10f : Spectral analysis**BMDP program library**

- P1T** Interactive or batch spectral analysis of one or two time series, with estimates of spectral density and coherence between variables. Options: print, plot (variable vs. time, lagged plots, complex demodulation, periodogram), handle missing values, remove seasonal means and linear trend, filtering, save results.

CMLIB subprogram library (MAXENTROPY sublibrary)

- BURG** Computes the coefficients of a finite length causal forward or backward prediction filter and uses both the forward and backward predictions in a symmetric manner to generate the maximum entropy spectrum by means of a Toeplitz recursion.

IMSL subprogram library

- FTFPS** Fast Fourier transform estimates of power spectra and cross spectra of time series.
- FTFREQ** Single or multichannel time series analysis in the time and frequency domains.

NAG subprogram library

- G13CAE** Calculates the smoothed sample spectrum of a univariate time series using one of four windows - rectangular, Bartlett, Tukey, or Parzen window. Double precision version is G13CAF.
- G13CBE** Calculates the smoothed sample spectrum of a univariate time series using spectral smoothing by the trapezium frequency (Danieli) window. Double precision version is G13CBF.
- G13CCE** Calculates the smoothed sample cross spectrum of a bivariate time series using one of four lag windows - rectangular, Bartlett, Tukey, or Parzen window. Double precision version is G13CCF.
- G13CDE** Calculates the smoothed sample cross spectrum of a bivariate time series using spectral smoothing by the trapezium frequency (Danieli) window. Double precision version is G13CDF.
- G13CEE** For a bivariate time series, calculates the cross amplitude spectrum and squared coherency, together with lower and upper bounds from the univariate and bivariate (cross) spectra. Double precision version is G13CEF.
- G13CFE** For a bivariate time series, calculates the gain and phase together with lower and upper bounds from the univariate and bivariate spectra. Double precision version is G13CFF.
- G13CGE** For a bivariate time series, calculates the noise spectrum together with multiplying factors for the bounds and the impulse response function and its standard error, from the univariate and bivariate spectra. Double precision version is G13CGF.

SPECTRLAN program library

- SPECAN** An interactive spectral analysis package for time series data. Produces periodograms, cumulative periodograms, continuous Fourier power spectra, cumulative power spectra, Fourier amplitude spectra, maximum entropy spectra, and integrated maximum entropy spectra. With DISSPLA graphics.

STATLIB subprogram library

- ACSPEC** Computes the series autospectrum from the Fourier transform of the user-supplied autocorrelation function, with user-supplied lag window truncation values.
- ASPEC** Computes the series autospectrum from the Fourier transform of the autocorrelation function, using the Jenkins and Watts window closing technique.
- ASPECS** Computes the series autospectrum from the Fourier transform of the autocorrelation function, with user-supplied lag window truncation values.
- IASPEC** Computes the integrated sample periodogram of a series (not recommended for long series because of the algorithm used).

L10g :	Cross-correlation analysis
---------------	----------------------------

L10g1 :	Parameter estimation
----------------	----------------------

IMSL subprogram library

FTCROS Means, variances, cross-covariances, and cross-correlations for two mutually stationary n channel time series.

FTCRXY Cross-covariance of two mutually stationary time series.

FTFPS Fast Fourier transform estimates of power spectra and cross spectra of time series.

FTTR Parameter estimates for a univariate transfer function model.

MINITAB interactive system

CCF Computes and graphs cross-correlations between two time series.

NAG subprogram library

G13BCE Calculates cross correlations between two time series. Double precision version is G13BCF.

STATLIB subprogram library

CCORRS Perform cross-correlation analysis between a pair of series.

CCSPEC Computes phase and squared coherency spectra from the Fourier transform of the user-supplied correlation functions, with user-supplied lag window truncation values.

CSPEC Computes phase and squared coherency spectra from the Fourier transform of the correlation functions, using the Jenkins and Watts window closing techniques.

CSPECS Computes phase and squared coherency spectra from the Fourier transform of the correlation functions, with user-supplied lag window truncation values.

ICSPEC Displays plots of the integrated sample phase and co-spectra for a pair of series (not recommended for long series because of the algorithm used).

L10g2 :	Forecasting
----------------	-------------

IMSL subprogram library

FTWENM Multichannel Wiener forecast.

FTWENX Maximum likelihood parameter estimates for a multichannel, single output time series model.

L11: Correlation Analysis

Correlation analysis deals with the relationships among one or more variables or sets of variables. Parametric analyses make assumptions about the distributions of the variables, while distribution-free (nonparametric) analyses make fewer or less restrictive assumptions.

Chapter L11 contains software for more sophisticated correlation analyses than the software in chapters L1e (for calculating parametric and nonparametric correlations) and chapters L4b and L4d (for calculation, elementary inference, and hypothesis testing of correlation statistics).

Consult the references listed in the introduction to chapter L for more information about correlation analysis.

January 1984

L11 : Correlation analysis**BMDP program library**

- P6M** Computes canonical correlation analysis for two sets of variables and Bartlett's test for the significance of the remaining eigenvalues, with printing. Options: weights, form of input, additional printing and plotting, save results.
- P6R** Computes the partial correlations of a set of variables after removing the linear effects of a second set of variables. Can be used for regression, especially if multiple dependent variables are present. Prints standard results. Options: weights, form of input, additional printing and plots.

STATLIB subprogram library

- COREL** Performs correlation analysis of a multivariate random sample.
- CORELS** Performs correlation analysis of a multivariate random sample with computed results returned to the user.

L12: Discriminant Analysis

Discriminant analysis is concerned with distinguishing among two or more populations. A "training" data set may be used to develop a discriminant function (commonly linear in the observational data) which is then used with data whose population membership is unknown. Classical discriminant analysis assumes that each population is multivariate normal with known variance-covariance structure, and substantial statistical inference is possible in this case. Stepwise techniques are commonly used to select a discriminant function. Both statistical and graphical techniques are useful in evaluating results.

This chapter contains one BMDP program and two IMSL subroutines; additional discriminant analysis software exists, especially of the distribution-free (nonparametric) form appropriate when the classical assumptions are inappropriate, and that software will be included in future editions.

January 1984

References

1. Anderson, T.W. (1958). *An Introduction to Multivariate Statistical Analysis*, John Wiley & Sons, New York.
2. Cacoullos, T. (Editor) (1973). *Discriminant Analysis and Applications*, Academic Press, New York.
3. Duda, R.O., and P.E. Hart (1973). *Pattern Classification and Scene Analysis*, Wiley-Interscience, New York.

L12 : Discriminant analysis**BMDP program library**

- P7M** Stepwise forward or backward discriminant analysis (including jackknifed classifications, percent correctly classified, Mahalanobis' distances, canonical variable coefficients, eigenvalues, scores, and plot of first two canonical variables). Options: interactive stepping, save results.

IMSL subprogram library

- ODFISH** Linear discriminant analysis method of Fisher for reducing the number of variables.
- ODNORM** Multivariate normal linear discriminant analysis among several known groups.

L13: Factor Analysis

Factor analysis is a statistical procedure used to analyze multivariate data. Factor analysis itself is commonly used in the social sciences. Principal components analysis, the most widely used special case of factor analysis, is commonly used with high dimensional data, and yields lower dimensional data which approximate the original data. Software for performing regression analysis using principal components as independent variables is classified at class L8a4ald.

January 1984

References

1. Harman, H.H. (1976). *Modern Factor Analysis*, The University of Chicago Press, Chicago.
2. Morrison, D.F. (1976). *Multivariate Statistical Methods*, McGraw-Hill, New York.
3. Timm, N.H. (1975). *Multivariate Analysis with Applications in Education and Psychology*, Brooks/Cole, Monterey, California.

L13 :	Factor analysis
--------------	------------------------

BMDP program library

- P4M** Provides four methods of initial factor extraction from a correlation or covariance matrix, and several methods of rotation, prints shaded correlations, factor loadings, factor score coefficients, factor scores, Mahalanobis distances, and plots. Options: weights, form of input, save results.
- P8M** Boolean factor analysis of binary (dichotomous) data. Options: initial estimates of the loading matrix, printing, save results.

IMSL subprogram library

- OFCOEF** Compute a matrix of factor score coefficients for input to IMSL routine OFSCOR.
- OFCOMM** Compute an unrotated factor loading matrix according to a common factor model by unweighted or generalized least sqrs., or by max. likelihood procedures.
- OFHARR** Transformation of unrotated factor loading matrix to oblique axes by Harris-Kaiser method.
- OFIMAG** Compute an unrotated factor loading matrix according to an image model.
- OFPROT** Oblique transformation of the factor loading matrix using a target matrix, including pivot and power vector options.
- OFRESI** Communalities and normalized factor residual correlation matrix calculation.
- OFROTA** Orthogonal rotation of a factor loading matrix using a generalized orthomax criterion, including quartimax, varimax, and equamax.
- OFSCHN** Orthogonal transformation of the factor loading matrix using a target matrix.
- OFSCOR** Compute a set of factor scores given the factor score coefficient matrix.

L13a :	Principal components analysis
---------------	--------------------------------------

IMSL subprogram library

- OFPRI** Compute an unrotated factor loading matrix according to a principal component model.
- OPRINC** Principal components of a multivariate sample of observations.

L14: Cluster Analysis

Cluster analysis is helpful in discovering structure in a large set of multivariate observations. Observations are clustered into groups such that observations in the same cluster are similar according to some criterion.

Starting from a matrix whose entries are measures of similarity or dissimilarity (e.g., Euclidean distance) between pairs of observations, one form of cluster analysis merges observations or clusters into larger clusters (using some measure such as the average distance between observations in two clusters) until some stopping criterion is met. Other types of cluster analysis divide the observations into clusters, produce overlapping clusters, and have constraints.

Future editions of GAMS will contain additional cluster analysis software.

January 1984

References

1. Hartigan, J.A. (1975). *Clustering Algorithms*, John Wiley & Sons, New York.
2. Jardine, N., and R. Sibson (1971). *Mathematical Taxonomy*, John Wiley & Sons, New York.

L14 :	Cluster analysis
--------------	------------------

L14a :	Unconstrained
---------------	---------------

L14a1 :	Nested
----------------	--------

L14a1a :	Joining (e.g., single link)
-----------------	-----------------------------

BMDP program library

P1M Stepwise cluster analysis of variables using one of four measures of similarity, three criteria for combining clusters, with printing of a summary table of clusters, shaded distance measure display, and a tree showing cluster formation. Options: form of input, additional printing and display.

P2M Stepwise cluster analysis of cases (observations) using one of four distance measures (including Euclidean and one for data that are frequency counts) and three linkage algorithms (single, centroid, k nearest neighbors), with a summary table of clusters and a cluster tree. Options: weights, standardized data.

IMSL subprogram library

OCDIS Pairwise Euclidean distances between the columns of a matrix.

OCLINK Perform a single-linkage or complete-linkage hierarchical cluster analysis given a similarity matrix.

L14a1b :	Divisive
-----------------	----------

L14a2 :	Non-nested
----------------	------------

BMDP program library

P3M Forms blocks (submatrices of the data matrix) where a subset of the cases (for a subset of the variables) cluster together, with printing of the blocks and tree diagrams for cases and for variables – appropriate for categorical data with few levels.

PKM By k-means procedure, partitions a set of cases (observations) into k clusters – beginning with user-specified initial clusters or one cluster, proceeding in divisive stepwise manner, then doing iterative reallocation – prints cluster profile and plot. Options: weights, standardize data (four ways), save results.

L14b :	Constrained
---------------	-------------

L14b1 :	One-dimensional
----------------	-----------------

L14b2 :	Two-dimensional
----------------	-----------------

L14c : Display

IMSL subprogram library

USTREE Print a binary tree (which may represent the output of a clustering algorithm in chapter O).**L15: Life Testing, Survival Analysis**

Two BMDP programs and one IMSL subroutine provide software for analysis of survival data. Other software available in the biomedical and the physical sciences communities will be included in future editions.

January 1984

L15 : Life testing, survival analysis

BMDP program library

P1L Estimates survival (time-to-response) distribution of patients who have been observed over varying periods of time by product-limit (Kaplan-Meier) or actuarial life table (Cutler-Ederer) method. Options: three forms of input, Mantel-Cox and Breslow test of equality of survival curves, five plots.

P2L Analyzes survival data with covariates using Cox proportional hazard regression model. Options: two forms of input, stepwise selection of covariates, time-dependent covariates, stratification, significance tests, three plots, print survival functions and residuals.

IMSL subprogram library

CLIFE Life table analysis.

M: Simulation, Stochastic Modelling

In order to build realistic mathematical models one must often account for the possibility of randomness. This is because elements of the phenomena under study often cannot be predicted in advance, but rather exhibit a probabilistic behavior which should be accounted for in the model. Probability models can be used to address a variety of questions such as "How many cabs should be on the street?", "How much time should pass before scheduling maintenance?", "How many beds should a hospital provide?", or "When should I replace my machines?". These questions share the common characteristic noted above—certain elements on which the answers depend can only be predicted in a statistical sense.

Software which can be used to build and to study stochastic models is classified in this chapter. A variety of such software exists, and will be cataloged in future editions of GAMS.

January 1984

M :	Simulation, stochastic modeling (<i>search also classes L6, L10</i>)
------------	--

M1: Simulation

Simulation models can be employed at various levels: (a) as explanatory devices to define a system or problem, (b) as analysis tools to determine critical elements, and (c) as predictors to aid in planning. In order to effectively utilize a simulation model however, one must understand the system under study, and focus on a specific problem which needs to be resolved.

In order to develop a simulation model, one must have an organized structure for viewing systems. A simulation language provides such a vehicle. It also has the ability to translate the model description into a form acceptable by a computing system. The computer can then be used to exercise the model, providing data that can be analysed in order to resolve the question under study.

January 1984

References

1. Pritsker, A. A. B. and C. D. Pegden (1979). *Introduction to Simulation and SLAM*, John Wiley & Sons, New York.

M1 :	Simulation
-------------	------------

M1a :	Discrete
--------------	----------

M1b :	Continuous (Markov models)
--------------	----------------------------

M2 :	Queueing
-------------	----------

M3 :	Reliability
-------------	-------------

M3a :	Quality control
--------------	-----------------

M3b :	Electrical network
--------------	--------------------

M4 :	Project optimization (e.g., PERT)
-------------	-----------------------------------

N: Data Handling

Software for data handling provides the basic tools needed to manipulate data structures. A data structure is a method for organizing data, together with a set of admissible operations on the data. While the most familiar data structures include real numbers, arrays, and sequential files, many more exotic data types have proven useful. Examples of these are character strings, hash tables, lists, stacks, trees, heaps, queues, and many types of files. The most common types of operations on complex data structures are input, output, insertion, deletion, searching, and sorting.

The input/output software cataloged here provides capabilities for the input and output of data types not usually available as standard options in programming languages. These capabilities include free-format input and easy-to-read output of vectors and matrices.

The data structures handled by software cataloged here are sequences and stacks. A sequence is a simple list, usually stored as an array. A stack is a list in which items are added or deleted (and retrieved) on a last-in first-out basis. The subroutines for stack management are from the PORT library; while these routines are principally used by other PORT routines (and are invisible to the user), they may be useful in their own right.

Software for sorting is described in the text for class N6 below.

January 1984

REFERENCES

1. Horowitz, E. and S. Sahni (1982). *Fundamentals of Data Structures*, Computer Sciences Press, Rockville, Maryland.
2. Knuth, D. E. (1973). *The Art of Computer Programming Volume 1: Fundamental Algorithms*, Addison-Wesley, Reading, Massachusetts.

N :	Data handling (search also class L2)
-----	--------------------------------------

N1 :	Input, output
------	---------------

DATA PAC subprogram library

READ	Performs a format-free read.
READG	Performs a format-free read of data from input unit = IRD.
SKIPR	Reads through (skips over) NLHEAD lines from input unit = 5.
WRITE	Writes out the contents of the vector X in an orderly and neat fashion.

IMSL subprogram library

USCWV	Print a complex vector.
USWBM	Print a matrix stored in band storage mode.
USWBS	Print a matrix stored in band symmetric storage mode.
USWCH	Print a complex matrix stored in Hermitian storage mode.
USWCM	Print a complex matrix stored in full storage mode.
USWFM	Print a matrix stored in full storage mode.
USWFBV	Print a vector.
USWSM	Print a matrix stored in symmetric storage mode.

N2 :	Bit manipulation
------	------------------

IMSL subprogram library

GTPBC	Count of the number of zero bits in a given subset of a real word.
--------------	--

N3 : Character manipulation**BMDP program library**

P4D Counts frequency of each number, letter, or symbol in single-column fields (A1 format). Options: input case label variables in A4 format, diagnostic printing useful in preliminary data screening. Specified characters may be replaced by blanks or symbols.

N4 : Storage management (e.g., stacks, heaps, trees)**IMSL subprogram library**

USTREE Print a binary tree (which may represent the output of a clustering algorithm in chapter O).

PORT subprogram library

ISTKGT Allocates (gets) an array from the storage stack for PORT library programs.
ISTKIN Initialize the length of the dynamic storage stack for PORT library programs.
ISTKMD Changes size of last stack allocation for PORT library programs.
ISTKQU Returns the number of available items that remain in the stack for PORT library programs.
ISTKRL Releases the last storage allocations requested for PORT library programs.
ISTKST Returns information on the status of the stack for PORT library programs.

N5 : Searching**N5a :** Extreme value**PORT subprogram library**

EXTRMI Finds extremal points of an integer function defined on a mesh.
EXTRMR Finds extremal points of a real function defined on a mesh. Double precision version is EXTRMD.
INTRVI Finds the interval in an integer array to which an integer element belongs.

N5b : Insertion position**PORT subprogram library**

INTRVR Finds the interval in a real array to which a real element belongs. Double precision version is INTRVD.

N5c : On a key**N6: Sorting**

Sorting is the rearrangement of units of data (called records) so that a particular data field (the key) is in ascending or descending order. Although more common in business processing, this problem has also seen many scientific applications since Von Neumann coded sorting algorithms on the EDVAC in 1945. Techniques for sorting can be generally classified into two types—internal or external—depending upon whether all the data fits in memory or not.

Internal sorting programs usually accept a one-dimensional array of keys and one or more "parallel" data arrays. The output of the program is either a set of rearranged data arrays ("active sorting") or a permutation array P ("passive sorting"). In the latter case the data is not reordered, but instead one may find the i^{th} element in sorted order in the $P(i)^{\text{th}}$ position. Good general purpose algorithms are those that minimize data comparisons and exchanges and do not have large memory requirements. Such programs usually run in time proportional to $n \log n$, where n is the number of records. C. A. R. Hoare's QUICKSORT is generally considered the best. These considerations may change for machines with special architecture, however.

External sorting methods are usually combinations of internal sorting and external merges, and here one is more interested in minimizing the amount of time spent in input or output of data. Most large computer manufacturers (those supporting COBOL, for instance), provide sorting programs as part of the operating system, although not all will interface with Fortran.

January 1984

References

1. Knuth, D. E. (1973). *The Art of Computer Programming Volume 3 : Sorting and Searching*, Addison-Wesley, Reading, Massachusetts.
2. Wirth, N. L. (1978). *Algorithms + Data Structures = Programs*, Prentice-Hall, Englewood Cliffs, New Jersey.

N6 :	Sorting
------	---------

N6a :	Internal
-------	----------

N6a1 :	Passive (i.e. construct pointer array, rank)
--------	--

N6a1a :	Integer
---------	---------

NAG subprogram library

M01ACE	Passively sort an integer vector into ascending order. (Identical to M01ACF.). Double precision version is M01ACF.
M01ADE	Passively sort an integer vector into descending order. (Identical to M01ADF.). Double precision version is M01ADF.

PORT subprogram library

SRTPAI	Passively sorts integer data into ascending order.
SRTPDI	Passively sorts integer data into descending order.

N6a1b :	Real
---------	------

N6a1b1 :	Single precision
----------	------------------

DATAPAC subprogram library

CODE	Codes the elements of the input vector X - 1.0 for minimum, 2.0 for next larger, etc.
RANK	Ranks (in ascending order) the N elements of the single precision vector X , and puts the resulting N ranks into the vector XR .

IMSL subprogram library

NMRANK	Numerical ranking.
--------	--------------------

VSAR Sorting of matrices (with options).

MINITAB interactive system

RANK Ranks the values in a vector. Ties are assigned the average rank.

NAG subprogram library

M01AAE Passively sort a real vector into ascending order. Double precision version is M01AAF.

M01ABE Passively sort a real vector into descending order. Double precision version is M01ABF.

PORT subprogram library

SRTPAR Passively sorts real data into ascending order. Double precision version is SRTPAD.

SRTPDR Passively sorts real data into descending order. Double precision version is SRTPDD.

N6a1b2 : Double precision

N6a1c : Character

PORT subprogram library

SRTPAH Passively sorts Hollerith data into ascending order.

SRTPDH Passively sorts Hollerith data into descending order.

N6a2 : Active

N6a2a : Integer

CMLIB subprogram library (SSORT sublibrary)

ISORT Sorts an integer array in either increasing or decreasing order. Optionally another integer array can be carried along.

NAG subprogram library

M01AGE Actively sort the rows of an integer matrix into ascending order of an index column. (Identical to M01AGF.). Double precision version is M01AGF.

M01AHE Actively sort the rows of an integer matrix into descending order of an index column. (Identical to M01AHF.). Double precision version is M01AHF.

M01ALE Actively sort an integer vector into ascending order of an index column and provide an index to the original order. (Identical to M01ALF.). Double precision version is M01ALF.

M01AME Actively sort an integer vector into descending order of an index column and provide an index to the original order. (Identical to M01AMF.). Double precision version is M01AMF.

M01AQE Actively sort an integer vector into ascending order (Singleton's implementation of Quicksort). (Identical to M01AQF.). Double precision version is M01AQF.

M01ARE Actively sort an integer vector into descending order (Singleton's implementation of Quicksort). (Identical to M01ARF.). Double precision version is M01ARF.

PORT subprogram library

SRTAI Actively sorts integer data into ascending order.

SRTDI Actively sorts integer data into descending order.

N6a2b : Real

N6a2b1 : Single precision

CMLIB subprogram library (SSORT sublibrary)

SSORT Sorts an array X (of N real numbers) into increasing or decreasing order. An optional array Y is carried along with X.

DATAPAC subprogram library

SORT Sorts (in ascending order) the N elements of the vector X and puts the resulting N sorted values into the vector Y.

SORTC Sorts (in ascending order) the N elements of the vector X and rearranges the elements of the vector Y.

SORTP Sorts (in ascending order) the N elements of the vector X, puts the resulting N sorted values into the vector Y, and puts the position (in the original vector X) of each of the sorted values into the single precision vector XPOS.

IMSL subprogram library

VSAR Sorting of matrices (with options).

VSORA Sorting of columns of a real matrix into ascending order of keys in rows.

VSRTA Sorting of arrays by algebraic value.

VSRTM Sorting of arrays by absolute value.

VS RTP Sorting of arrays by absolute value permutations returned.

VS RTR Sorting of arrays by algebraic value permutations returned.

MINITAB interactive system

ORDER Sorts in ascending order the values in each of one or more vectors.

SORT Sorts a vector in ascending order and optionally carries along other vectors.

NAG subprogram library

M01AEE Actively sort the rows of a real matrix into ascending order of an index column. Double precision version is M01AEF.

M01AFE Actively sort the rows of a real matrix into descending order of an index column. Double precision version is M01AFF.

M01AJE Actively sort a real vector into ascending order and provide an index to the original order. Double precision version is M01AJF.

M01AKE Actively sort a real vector into descending order and provide an index to the original order. Double precision version is M01AKF.

M01ANE Actively sort a real vector into ascending order (Singleton's implementation of Quicksort). Double precision version is M01ANF.

M01APE Actively sort a real vector into descending order (Singleton's implementation of Quicksort). Double precision version is M01APF.

PORT subprogram library

SRTAR Actively sorts real data into ascending order. Double precision version is SRTAD.

SRTDR Actively sorts real data into descending order. Double precision version is SRTDD.

N6a2b2 : Double precision

IMSL subprogram library

VSODA Sorting of columns of a double precision matrix in ascending order of keys in rows.

N6a2c : Character

CMLIB subprogram library (SSORT sublibrary)

CSORT Sorts a character array in either increasing or decreasing order. Optionally another character array can be carried along.

NAG subprogram library

M01BAE Actively sort a character vector into reverse alphanumeric order (Singleton's implementation of Quicksort). (Identical to M01BAF.). Double precision version is M01BAF.

M01BBE Actively sort a character vector into alphanumeric order (Singleton's implementation of Quicksort). (Identical to M01BBF.). Double precision version is M01BBF.

M01BCE Actively sort the columns of a character matrix into reverse alphanumeric order of an index column. (Identical to M01BCF.). Double precision version is M01BCF.

M01BDE Actively sort the columns of a character matrix into alphanumeric order of an index column. (Identical to M01BDF.). Double precision version is M01BDF.

PORT subprogram library

SRTAH Actively sorts Hollerith data into ascending order.

SRTDH Actively sorts Hollerith data into descending order.

N6b : External

N7 : Merging

N8 : Permuting

IMSL subprogram library

VSRTU Interchange the rows or columns of a matrix using a permutation vector such as the one obtained from IMSL routines VS RTP or VS RTR.

PORT subprogram library

SRTRH Rearranges Hollerith data according to permutation stored in IP.

SRTRI Rearranges integer data according to permutation stored in IP.

SRTRR Rearranges real data according to permutation stored in IP. Double precision version is SRTRD.

O: Symbolic Computation

This chapter is designed to contain software for the manipulation of mathematical expressions in their natural, symbolic form rather than just the manipulation of numbers. Such programs have been in use for some time to perform tedious, but important, symbolic computations in a number of fields such as celestial mechanics and quantum electrodynamics. Many standard operations of algebra and the calculus, including polynomial and rational arithmetic, differentiation, integration, and taking limits can routinely be done in such systems.

Unfortunately, no software for symbolic computations is cataloged in the current edition of GAMS. Information about existing software can be found in the references.

January 1984

References

1. van Hulzen, J. A. and J. Calmet (1982). Computer Algebra Systems, in *Computer Algebra - Symbolic and Algebraic Computation*, B. Buchberger, G. E. Collins, and R. Loos (Editors), Springer-Verlag, New York.
2. Yun, D. Y. Y. and R. D. Stoutemyer (1980). Symbolic Mathematical Computation, in *Encyclopedia of Computer Science and Technology*, J. Belzer, A. G. Holzman, and A. Kent (Editors), Vol. 15, pp. 235-310.

P: Computational Geometry

Computational geometry is the design and analysis of algorithms for geometric computations. One example of a computational geometry problem is finding the convex hull of a set of n points in the Euclidean plane, i.e. the smallest convex set containing the points. While naive algorithms involve examining all pairs of points and require $O(n^2)$ operations, more efficient algorithms require $O(n \log n)$ operations, and the newest algorithms require, on average, $O(n)$ operations when the spatial distribution of the points is random. Since each point must be examined at least once, $O(n)$ is obviously the lower bound on the number of operations. The convex hull of a set of points can be used to efficiently find the circle of smallest radius containing all of the points; the center of the circle is the location which minimizes the maximum distance to any of the given points and would "be a suitable location for an emergency service facility if it were desired to minimize the worst-case response time" (Shamos, 1977). Other areas of computational geometry applications include statistics and graphics.

A substantial amount of computational geometry software has been written, and we hope to include more in future editions of GAMS.

January 1984

REFERENCES

1. Shamos, M. I. (1977). *Problems in Computational Geometry*.

P :	Computational geometry (search also classes G, Q)
------------	---

P1 :	One dimension
-------------	---------------

P2 :	Two dimensions
-------------	----------------

P2a :	Points, lines
--------------	---------------

P2a1 :	Relationships
---------------	---------------

P2a1a :	Closest and farthest points
----------------	-----------------------------

P2a1b :	Intersection
----------------	--------------

P2a2 :	Graph construction
---------------	--------------------

P2a2a :	Convex hull
----------------	-------------

P2a2b :	Minimum spanning tree
----------------	-----------------------

P2a2c :	Region partitioning
----------------	---------------------

P2a2c1 : Triangulation

NAG subprogram library

D03MAE Triangulation of a plane region. Double precision version is D03MAF.

P2a2c2 : Voronoi diagram

P2b : Polygons (e.g., intersection, hidden line problems)

P2c : Circles

P3 : Three dimensions

P3a : Points, lines, planes

P3b : Polytopes

P3c : Spheres

P4 : More than three dimensions

Q: Graphics

The simplest type of graphics output device imaginable—the line printer—is the only type of device supported by the software currently cataloged in this chapter. For statistical data analysis, this type of software is often sufficient. More sophisticated graphics software can be used to produce graphs and diagrams on graphics hardware devices such as drum or flatbed plotters, electrostatic printer/plotters, or CRT screens. This software typically has many internal routines written for specific hardware and thus is often difficult to transport from one machine to another.

Users may have software control of graphics devices at a number of different levels. For example, one Fortran subroutine call might simply draw a line between two specified points. Another might plot a smooth curve through a given set of points. Yet another might draw a two-dimensional perspective plot of a surface in three dimensions represented by a table of function values, complete with the title and axis labels.

The most frequently used line-printer plots are histograms for univariate data and scatter diagrams for bivariate data. Options for these plots include the axis limits, special plot characters, printing multiple plots on one set of axes, selection of a subset of the data, and the size of the plot. Software for line-printer plots of time series, pseudo-three dimensional plots, a plot of a binary tree, and EDA (experimental data analysis) graphics are also available.

Four presentation-quality graphics software products are available to NBS staff but are not cataloged in this edition of GAMS. They are DATAPLOT (Filliben, 1980), a Tektronix-based interactive system for graphical data analysis, DISSPLA (ISSCO, 1981), a library of Fortran subroutines allowing the user extensive software control, including color, PLOT10 (Tektronix, 1977), Tektronix's basic graphics Fortran subroutine package, and SURFACE II (Sampson, 1978), a high-level control language for plotting surfaces and contour maps and which handles data on irregular grids. DISSPLA and PLOT10 are available on both the Sperry 1100 in Gaithersburg and the Cyber 750 in Boulder, while DATAPLOT is only available on the Sperry 1100 and SURFACE II is only available on the Cyber 750.

January 1984

REFERENCES

1. Filliben, J. J. (1980). A Review of DATAPLOT – An Interactive High-Level Language for Graphics, Non-Linear Fitting, Data Analysis, and Mathematics, in *1980 Proceedings of the Statistical Computing Section*, American Statistical Association, Washington, D.C.
2. ISSCO (Integrated Software Systems Corporation) (1981). *DISSPLA User's Manual*, ISSCO, San Diego, California.
3. Sampson, R. J. (1978). *SURFACE II Graphics System*, Revision 1, Series on Spatial Analysis, Number 1, Kansas Geological Survey, Lawrence, Kansas.
4. Tektronix (1977). *PLOT10 Terminal Control System Users Manual*, Tektronix, Beaverton, Oregon.

Q : Graphics (search also classes L9, P)

Q1 : Line printer plotting

DATAPAC subprogram library

HIST	Produces 2 histograms (with differing class widths) of the data in the input vector X.
PLOT	Yields a one-page printer plot of Y(I) versus X(I).
PLOT10	Yields a one-page printer plot of Y(I) versus X(I) for a subset of the data, with special plot characters, and with specified axis limits and labels.
PLOT6	Yields a one-page printer plot of Y(I) versus X(I) for specified axis limits.
PLOT7	Yields a one-page printer plot of Y(I) versus X(I) with special plot characters and for specified axis limits.
PLOT8	Yields a one-page printer plot of Y(I) versus X(I) with special plot characters for a subset of the data with specified axis limits.
PLOT9	Yields a one-page printer plot of Y(I) versus X(I) with special plot characters and for specified axis limits and axis labels.
PLOT C	Yields a one-page printer plot of Y(I) versus X(I) with special plotting characters.
PLOT CT	Yields a narrow-width (71-character) plot of Y(I) versus X(I) with special plotting characters.

PLOTS	Yields a one-page printer plot of Y(I) versus X(I) for a subset of the data.
PLOTSC	Yields a one-page printer plot of Y(I) versus X(I) with special characters for a subset of the data.
PLOTST	Yields a narrow-width (71-character) of Y(I) versus X(I) for a subset of the data.
PLOTT	Yields a narrow-width (71-character) plot of Y(I) versus X(I).
PLOTX	Yields a one-page printer plot of X(I) versus I.
PLOTXT	Yields a narrow-width (71-character) plot of X(I) versus I.
PLTSCT	Yields a narrow-width (71-character) plot of Y(I) versus X(I) with special plot characters and a subset of the data.

IMSL subprogram library

USHHST	Print a horizontal histogram.
USHST	Print a vertical histogram.
USPLO	Printer plot of up to ten functions.
USPLOD	Printer plot of up to ten functions.
USTREE	Print a binary tree (which may represent the output of a clustering algorithm in chapter O).

MINITAB interactive system

HISTOGRAM	Prints a histogram of the values in each of one or more vectors, with optional user-specification of the first midpoint and the interval width.
L PLOT	Prints a letter plot with symbols corresponding to numerical "tag" values. Scale specification is optional.
M PLOT	Prints multiple scatter diagrams on the same axis.
PLOT	Prints a scatter diagram, with optional scale specification.
T PLOT	Prints pseudo three-dimensional plot of y versus x versus z, with symbols indicating the values of z, and with optional scale specification.
TS PLOT	Prints a scatter diagram of a time series, optionally using symbols modulo the period. Handles missing values.

NAG subprogram library

G01AGE	Line printer scatter plot of two variables. Double precision version is G01AGF.
---------------	---

STATLIB subprogram library

MPLT	Displays a 50x100 character line printer plot of several dependent variables vs. a common independent variable.
MPLTH	Displays a 50x50 character line printer plot of several dependent variables vs. a common independent variable.
MPLTHL	Displays a 50x50 character line printer plot of several dependent variables vs. a common independent variable, with user control of the plot limits.
MPLTL	Displays a 50x100 character line printer plot of several dependent variables vs. a common independent variable, with user control of the plot limits.
PLT	Displays a 50x100 character line printer scatter plot.
PLTH	Displays a 50x50 character line printer scatter plot.
PLTHL	Displays a 50x50 character line printer scatter plot with user control of plot limits.
PLTL	Displays a 50x100 character line printer scatter plot with user control of plot limits.
SPLT	Displays a 50x100 character line printer scatter plot with user control of the plotting symbol used for each point.
SPLTH	Displays a 50x50 character line printer scatter plot with user control of the plotting symbol used for each point.
SPLTHL	Displays a 50x50 character line printer scatter plot with user control of the plot limits and of the

plotting symbol used for each point.

- SPLTL** Displays a 50x100 character line printer scatter plot with user control of the plot limits and of the plotting symbol used for each point.
- VPLT** Displays an Nx100 character line printer plot of the N values of a series (horizontal axis) vs. their indices (vertical axis).
- VPLT2** Displays an Nx100 character line printer plot of the N values of each of two series (horizontal axis) vs. their indices (vertical axis).
- VPLT2L** Displays an Nx100 character line printer plot of the N values of each of two series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits.
- VPLTB** Displays an Nx100 character line printer bar plot of the N values of a series (horizontal axis) vs. their indices (vertical axis).
- VPLTBL** Displays an Nx100 character line printer bar plot of the N values of a series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits.
- VPLTL** Displays an Nx100 character line printer plot of the N values of a series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits.

R: Service Routines

This chapter contains subprograms which perform fairly low-level utility functions such as error checking, error handling, and retrieval of information about machine characteristics. These routines are generally not of immediate interest to routine users of software libraries. However, they are quite important for those writing software intended to be portable to and reliable on a wide variety of machines.

January 1984

R: Service routines

R1: Machine-dependent constants

Modules classified in this section are useful for increasing the portability of programs. In Fortran, some degree of portability can be obtained by using a subset of Fortran acceptable to all compilers and by isolating all machine-dependent parameters so they are easily changed when moving to a new environment. The latter can be done in two distinct ways. The most direct is to place machine specific information in a visible place in your program and document clearly which machine parameters you are using and how they should be changed to move to another computer, or perhaps to another precision. The advantage of this is that your program becomes fully self-contained. An alternate approach is to use widely available subroutines which return information about the current machine environment. Examples are the routines IIMACH, R1MACH and D1MACH in the CMLIB and PORT libraries, as well as the routines in chapters X01 and X02 of the NAG library. These allow you to write very portable code. When machine information is required it is obtained "automatically" by evaluation of these functions within your program.

The decision as to which of these approaches to use depends mostly on the volume of software that might have to be transported. The first seems quite simple for only one, or at most a few, routines. The second is very useful if a substantial amount of software is being moved, since all the changes are localized in the machine-constant routines. Even when moving a small block of code the latter is often better since it is easy to forget to make small but necessary changes in lower level routines.

January 1984

References

1. Brown, W. S. and S. I. Feldman (1980). Environment Parameters and Basic Functions for Floating-Point Computation, *ACM Transactions on Mathematical Software*, Vol. 6, No. 4, pp. 510-523.
2. Fox, P. A., A. D. Hall, and N. L. Schryer (1978). The PORT Mathematical Subroutine Library, *ACM Transactions on Mathematical Software*, Vol. 4, No. 2, pp. 104-126.

R1: Machine-dependent constants

CMLIB subprogram library (MACHCONST sublibrary)

- IIMACH** Provides integer machine dependent information, e.g. largest integer.
- R1MACH** Provides single precision machine dependent information, e.g. R1MACH(4) returns machine epsilon. Double precision version is D1MACH.

IMSL subprogram library

- UGETIO** To retrieve current values and to set new values for input and output unit identifiers.

NAG subprogram library

- X01AAE** Pi. Double precision version is X01AAF.
- X01ABE** Euler's constant, gamma. Double precision version is X01ABF.
- X02AAE** Smallest possible e such that $1.0+e > 1.0$. Double precision version is X02AAF.
- X02ABE** Smallest representable positive real number. Double precision version is X02ABF.

X02ACE	Largest representable positive real number. Double precision version is X02ACF.
X02ADE	Ratio of X02ABE to X02AAE. Double precision version is X02ADF.
X02AEE	Largest negative permissible argument for exp. Double precision version is X02AEF.
X02AFE	Returns the value of the largest positive argument permitted for EXP. Double precision version is X02AFF.
X02AGE	Smallest representable positive real number with representable inverse. Double precision version is X02AGF.
X02AHE	Returns the value of the largest positive argument permitted for SIN and COS. Double precision version is X02AHF.
X02BAE	Base of floating-point arithmetic. Double precision version is X02BAF.
X02BBE	Largest representable integer. Double precision version is X02BBF.
X02BCE	Largest positive integer power to which 2.0 can be raised without overflow. Double precision version is X02BCF.
X02BDE	Largest negative integer power to which 2.0 can be raised without underflow. Double precision version is X02BDF.
X02BEE	Maximum number of decimal digits that can be represented. Double precision version is X02BEF.
X02CAE	Estimate of active-set size (on machines with paged virtual store). Double precision version is X02CAF.
X02DAE	Switch for taking precautions to avoid underflow. Double precision version is X02DAF.

PORT subprogram library

IIMACH	Provides the integer constants required to adapt PORT library programs to individual computers.
R1MACH	Provides the single precision machine-dependent constants required to adapt PORT library programs to individual computers. Double precision version is D1MACH.

R2 :	Error checking (e.g., check monotonicity)
-------------	---

PORT subprogram library

MONOI	Test if an integer vector is monotone increasing or decreasing.
MONOR	Test if a real vector is monotone increasing or decreasing. Double precision version is MONOD.
SMONOI	Test if an integer vector is strictly monotone increasing or decreasing.
SMONOR	Test if a real vector is strictly monotone increasing or decreasing. Double precision version is SMONOD.

R3: Error handling

Most debugging aids for the casual programmer are very system dependent. Within large collections of programs such as the commercial libraries certain conventions have been established for reporting errors that occur. For the most part library routines do this by calling a subroutine which performs various actions depending on the severity of the error. Library users do not often need to know about these programs. Programmers who are developing a "package" for use by others may wish to take advantage of some of these utilities, however. One such error handler is XERROR, a public-domain package available in (and used by) CMLIB.

In some cases it is necessary for casual users to change some of the defaults in use by the error handler. For example, every error may cause the program to abort. This is easily changed, and the way that this is done depends upon the particular library in use. For detailed information about the error handlers in the IMSL, NAG, and PORT libraries, one should consult the appropriate library reference manual (see the Library Reference section of GAMS).

R3 : Error handling**R3a :** Set criteria for fatal errors

CMLIB subprogram library (XERROR sublibrary)

XSETF Set KONTRL for XERROR, default is =2.

IMSL subprogram library

UERSSET Set message level for IMSL routine UERTST.

PORT subprogram library

ENTSRC Saves current recovery mode status and sets a new one for PORT library programs.**RETSRC** Test and reset error recovery mode for PORT library programs.**R3b :** Set unit number for error messages

CMLIB subprogram library (XERROR sublibrary)

XSETUA Set up to 5 output unit numbers.**XSETUN** Set one output unit number.

IMSL subprogram library

UGETIO To retrieve current values and to set new values for input and output unit identifiers.

NAG subprogram library

X04AAE Return or set unit number for error messages for Nag library programs. Double precision version is X04AAF.**X04ABE** Return or set unit number for advisory messages for NAG library programs. Double precision version is X04ABF.**R3c :** Other utility programs

CMLIB subprogram library (XERROR sublibrary)

NUMXER Get most current message number.**XERABT** Terminate run and print traceback. (Requires system dependent programming to execute properly, else just STOPs.).**XERCLR** Clear current message number.**XERCTL** Perform special error processing of one message.**XERDMP** Print error summary and clear tables.**XERMAX** Set limit of MAX times each message can be printed.• **XERROR** Process a message.**XERRWV** Process a message with numeric values.**XGETF** Get current value of KONTRL.**XGETUA** Get current output unit numbers.**XGETUN** Get current output unit number.

IMSL subprogram library

UERTST Print a message reflecting an error condition.

NAG subprogram library

P01AAE Return value of error indicator, or terminate with an error message. Used exclusively by NAG library programs. Double precision version is P01AAF.

PORT subprogram library

ENTER Save current error recovery mode and storage allocation status for PORT library programs.

EPRINT Print the current error message if the program is in the error state for PORT library programs.

ERROFF Turns off the error state for PORT library programs.

LEAVE Restores prior error recovery mode and reset the stack for PORT library programs.

NERROR Provides the current error number for PORT library programs.

SETERR Sets the error indicator and depending on options prints a message and provides a dump for PORT library programs.

R4: Documentation retrieval

This section contains modules whose sole purpose is to provide information about programming conventions within a library or package. For example the UHELP routines give information about conventions in the IMSL library.

January 1984

R4 : Documentation retrieval**IMSL subprogram library**

UHELP Display methods of obtaining info on IMSL conventions regarding various subjects provide means for individual sites to supply users with site specific info.

UHELP1 Write information regarding IMSL conventions and notation to an output file.

UHELP2 Write information regarding IMSL input and output conventions.

UHELP3 Write information regarding IMSL error detecting facilities.

UHELP4 Write information regarding matrix/vector storage modes used in IMSL subroutines.

S: Software Development Tools

This chapter is designed to contain software tools which ease the process of program development and maintenance. The types of jobs performed by such tools are program transformation (e.g., convert to double precision), static analysis (e.g., flow analysis, interface analysis), and dynamic analysis (e.g., tracing, timing, assertion checking).

Unfortunately, no software development tools are cataloged in the current edition of GAMS. Information about existing software can be found in the reference.

January 1984

References

1. Houghton, R.C., Jr. (1983). Software Development Tools: A Profile, *Computer*, Vol. 16, No. 5, pp. 63-70.

S :	Software development tools
-----	----------------------------

S1 :	Program transformation
------	------------------------

S2 :	Static analysis
------	-----------------

S3 :	Dynamic analysis
------	------------------

MODULE DICTIONARY

The "module" is the smallest unit cataloged in GAMS. A module may be a Fortran subprogram, a stand-alone program, or a command in an interactive system. In this section we present a summary description of each module in alphabetical order. The following legend is an explanation of the information included in the summaries.

LEGEND

Entries in the Module Dictionary take the form

Name	Description Form Class(es) Usage On-line doc Tests Access See also
------	--

where

Name	is the module name.
-------------	---------------------

Description	briefly describes the purpose of the module.
--------------------	--

Form	indicates (a) whether the module is portable or proprietary, (b) whether the module is a subprogram, a stand-alone program, or a command in an interactive system, and (c) the name of the library (and sublibrary, if appropriate) containing the module. The use of proprietary software is governed by a licensing agreement, while the use of software designated as portable is unrestricted. (See the legend of the Library Reference, page D0, for further discussion of portability issues.) If the module is a subprogram then the language in which it is coded and the arithmetic precision (single or double) of the computed results is also given.
-------------	--

Class(es)	lists up to three GAMS classifications which best describe the purpose of the module.
------------------	---

Usage	indicates how to use the module. This is the call sequence for subprograms and the syntax of the command for interactive systems.
--------------	---

On-line doc	indicates how to obtain detailed on-line documentation using the Sperry 1100 system at NBS. This usually takes the form "CTS command (EXEC command)."
--------------------	---

Tests	occurs (optionally) only for subprograms. It gives the name of the file element on the Sperry 1100 system at NBS which contains a main program which calls the module. This may be either an example program or a program used to test the integrity of the subprogram.
--------------	---

Access	indicates how to gain access to the module on the Sperry 1100 system at NBS. For subprograms this is the name of a relocatable library, for stand-alone programs it is the command used to execute the program, and for interactive system commands it is the command used to execute the system.
---------------	---

See also	(optional) gives the names of other modules in the same library which are almost always used in conjunction with this module.
-----------------	---

A

- A02AAE** Evaluates the square root of a complex number. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is A02AAF. | Class(es): C2 | Usage: CALL A02AAE(XR,XI,YR,YI) | On-line doc: CALL GAMSDOC A02AAE (or @PRT NAG*DOC.A02AAE) | Access: LIB NBS*NAG
- A02AAF** Evaluates the square root of a complex number. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is A02AAE. | Class(es): C2 | Usage: CALL A02AAF (XR, XI, YR, YI) | On-line doc: CALL GAMSDOC A02AAF (or @PRT NAG*DOC.A02AAF) | Access: LIB NBS*NAG
- A02ABE** Modulus of a complex number. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is A02ABF. | Class(es): A4a | Usage: D = A02ABE (XR, XI) | On-line doc: CALL GAMSDOC A02ABE (or @PRT NAG*DOC.A02ABE) | Access: LIB NBS*NAG
- A02ABF** Modulus of a complex number. | Proprietary double precision Fortran subprogram in NAG library. | Class(es): A4b | Usage: D = A02ABF (XR, XI) | On-line doc: CALL GAMSDOC A02ABF (or @PRT NAG*DOC.A02ABF) | Access: LIB NBS*NAG
- A02ACE** Quotient of two complex numbers. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is A02ACF. | Class(es): A4a | Usage: CALL A02ACE (XR, XI, YR, YI, ZR, ZI) | On-line doc: CALL GAMSDOC A02ACE (or @PRT NAG*DOC.A02ACE) | Access: LIB NBS*NAG
- A02ACF** Quotient of two complex numbers. | Proprietary double precision Fortran subprogram in NAG library. | Class(es): A4b | Usage: CALL A02ACF (XR, XI, YR, YI, ZR, ZI) | On-line doc: CALL GAMSDOC A02ACF (or @PRT NAG*DOC.A02ACF) | Access: LIB NBS*NAG
- A10GII** Solves generalized eigenvalue problem by inverse iteration via Gauss elimination. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): D4b2 | Usage: CALL A10GII(W,NW,N,N2,X,Y,SHIFT,EPS,LIMIT,EV,IPR) | On-line doc: @PRT,S MATHWARE*NASHLIB.A10GII | Tests: MATHWARE*NASHLIB.A10 | Access: See individual sublibrary documentation
- A11VS** Standardizes a complex vector to have maximum component of magnitude equal to one. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): D1a6 | Usage: CALL A11VS(N,X,Y,VNORM) | On-line doc: @PRT,S MATHWARE*NASHLIB.A11VS | Tests: MATHWARE*NASHLIB.A11-12 | Access: See individual sublibrary documentation
- A12CVR** Residuals for a complex eigenvalue $c+if$ and eigenvector $x+iy$ of the matrix $A+iz$. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): D4c | Usage: CALL A12CVR(N,X,Y,A,NA,Z,NZ,E,F,U,V) | On-line doc: @PRT,S MATHWARE*NASHLIB.A12CVR | Tests: MATHWARE*NASHLIB.A11-12 | Access: See individual sublibrary documentation
- A13ESV** Solves eigenproblem for real symmetric matrix via Singular Value Decomposition. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): D4a1 | Usage: CALL A13ESV(N,A,NA,EPS,H,ISWP,IPR,Z) | On-line doc: @PRT,S MATHWARE*NASHLIB.A13ESV | Tests: MATHWARE*NASHLIB.A13 | Access: See individual sublibrary documentation
- A14JE** Jacobi algorithm for the eigenvalues and eigenvectors of a real symmetric matrix. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): D4a1 | Usage: CALL A14JE(N,A,NA,V,NV,ISWP,IPR,SETV,COMV) | On-line doc: @PRT,S MATHWARE*NASHLIB.A14JE | Tests: MATHWARE*NASHLIB.A14-15 | Access: See individual sublibrary documentation
- A15GSE** Solution of the generalized symmetric eigenvalue problem by two applications of the Jacobi algorithm. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): D4b1 | Usage: CALL A15GSE(N,A,NA,B,NB,V,NV,FAIL,ISWP,IPR) | On-line doc: @PRT,S MATHWARE*NASHLIB.A15GSE | Tests: MATHWARE*NASHLIB.A14-15 | Access: See individual sublibrary documentation
- A16GS** Grid or equal interval search along a line. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): G1a2 | Usage: CALL A16GS(U,V,N,FNS,IFN,TOL,IPR,T,VAL) | On-line doc: @PRT,S MATHWARE*NASHLIB.A16GS | Tests: MATHWARE*NASHLIB.A16 | Access: See individual sublibrary documentation
- A17LS** Success failure linear search with parabolic inverse interpolation. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): G1a2 | Usage: CALL A17LS(B,ST,FUNS,IFN,NOCOM,IPR) | On-line doc: @PRT,S MATHWARE*NASHLIB.A17LS | Tests: MATHWARE*NASHLIB.A17 | Access: See individual sublibrary documentation
- A18RF** Root finding or minimization by bisection and false position. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): G1a2 F1b | Usage: CALL U,V,COUNT,NBIS,FUN,TOL,NOCOM,IPR) | On-line doc: @PRT,S MATHWARE*NASHLIB.A18RF | Tests: MATHWARE*NASHLIB.A18 | Access: See individual sublibrary documentation
- A19NM** Nelder Mead simplex function minimization. A short routine, but often quite successful. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): G1b2 | Usage: CALL A19NM(N,B,X,NX,N2,NOCOM,IFN,VL,FUN,STEP,IPR) | On-line doc: @PRT,S MATHWARE*NASHLIB.A19NM | Tests: MATHWARE*NASHLIB.A19-20 | Access: See individual sublibrary documentation
- A1SVD** Singular Value Decomposition by means of orthogonalizing plane rotations. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): D6 | Usage: CALL A1SVD(M,N,A,NA,EPS,V,NV,Z,IPR) | On-line doc: @PRT,S MATHWARE*NASHLIB.A1SVD | Tests: MATHWARE*NASHLIB.A1-2 | Access: See individual sublibrary documentation
- A21VM** Variable metric minimization method. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): G1b1b | Usage: CALL A21VM(N,B,BH,NBH,X,C,G,T,IFN,IG,NOCOM,IPR,P0,FUN,DER) | On-line doc: @PRT,S

- MATHWARE*NASHLIB.A21VM | Tests: MATHWARE*NASHLIB.A21 | Access: See individual sublibrary documentation
- A22CGM** Function minimization by conjugate gradients. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): G1b1b | Usage: CALL A22CGM(N,B,FUN,DER,NOCOM,IPR,IFN,IG,EPS,G,X,T,C,P0) | On-line doc: @PRT,S MATHWARE*NASHLIB.A22CGM | Tests: MATHWARE*NASHLIB.A22 | Access: See individual sublibrary documentation
- A23MRT** Modified Marquardt procedure for minimizing a nonlinear sum of squares function. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): K1b1a2 | Usage: CALL A23MRT(N,B,M,TOL,A,C,N2,X,V,D,RES,DRES,NOCOM,P0,IFN,IG,F,IPR) | On-line doc: @PRT,S MATHWARE*NASHLIB.A23MRT | Tests: MATHWARE*NASHLIB.A23 | Access: See individual sublibrary documentation
- A24CG** Solution of a consistent system of linear equations with symmetric non negative definite coefficient matrix. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): D2b1b D2b4 | Usage: CALL A24CG(N,B,C,TOL,G,IPR,APR,V,T,IMULT) | On-line doc: @PRT,S MATHWARE*NASHLIB.A24CG | Tests: MATHWARE*NASHLIB.A24 | Access: See individual sublibrary documentation
- A25RQM** Rayleigh quotient minimization by conjugate gradients. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): D4b1 | Usage: CALL A25RQM(N,X,EPS,APR,BPR,KPR,Y,Z,T,G,A,B,P0,IPR) | On-line doc: @PRT,S MATHWARE*NASHLIB.A25RQM | Tests: MATHWARE*NASHLIB.A25 | Access: See individual sublibrary documentation
- A2LSVD** Least squares solution of rectangular linear system by Singular Value Decomposition. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): D0 | Usage: CALL A2LSVD(M,N,A,NA,EPS,V,NV,Z,IPR,Y,G,X,Q,ESVD,NTOL) | On-line doc: @PRT,S MATHWARE*NASHLIB.A2LSVD | Tests: MATHWARE*NASHLIB.A1-2 | Access: See individual sublibrary documentation
- A3GR** Given's reduction of a real rectangular matrix. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): D5 | Usage: CALL A3GR(M,N,A,NA,Q,NQ,EPS,SAVEQ) | On-line doc: @PRT,S MATHWARE*NASHLIB.A3GR | Tests: MATHWARE*NASHLIB.A3 | Access: See individual sublibrary documentation
- A5GE** Gauss elimination with partial pivoting for solution of system of linear equations, $Ax=f$. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): D2a1 | Usage: CALL A5GE(A,NA,N,NP,D,TOL) | On-line doc: @PRT,S MATHWARE*NASHLIB.A5GE | Tests: MATHWARE*NASHLIB.A5-6 | Access: See individual sublibrary documentation
- A6BS** Back substitution for the solution of a triangular system of linear equations, $Rx=f$. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): D2a1 | Usage: CALL A6BS(A,NA,N,NP) | On-line doc: @PRT,S MATHWARE*NASHLIB.A6BS | Tests: MATHWARE*NASHLIB.A5-6 | Access: See individual sublibrary documentation | See also: A5GE
- A7CH** Cholesky decomposition of symmetric non-negative definite matrix in compact storage. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): D2b1b | Usage: CALL A7CH(A,N2,N,INDEF) | On-line doc: @PRT,S MATHWARE*NASHLIB.A7CH | Tests: MATHWARE*NASHLIB.A7-8 | Access: See individual sublibrary documentation
- A8CS** Cholesky back substitution for the solution of consistent sets of linear equations with symmetric coefficient matrices, compact storage. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): D2b1b | Usage: CALL A8CS(A,N2,X,N) | On-line doc: @PRT,S MATHWARE*NASHLIB.A8CS | Tests: MATHWARE*NASHLIB.A7-8 | Access: See individual sublibrary documentation | See also: A7CH
- A9GJ** Bauer Reinsch inversion of a positive definite symmetric matrix by a modification of Gauss-Jordan method. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): D2b1b | Usage: CALL A9GJ(A,N2,N,INDEF,X) | On-line doc: @PRT,S MATHWARE*NASHLIB.A9GJ | Tests: MATHWARE*NASHLIB.A9 | Access: See individual sublibrary documentation
- ABIBN** Analysis of balanced incomplete block and balanced lattice designs. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a2b1 | Usage: CALL ABIBN (ISW,Y,N,IND,IPOT,EM,BM,S,SADJ,NDF,EF,IER) | On-line doc: CALL GAMSDOC ABIBN (or @PRT IMSL*DOC.ABIBN) | Access: LIB NBS*IMSL
- ACF** Computes and graphs the autocorrelations of a time series, and optionally saves results. | Command in MINITAB Proprietary interactive system. | Class(es): L10c | Usage: ACF [with up to K lags] for series in C [put results in C] | On-line doc: HELP ACF (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- ACORR** Performs autocorrelation analysis of a series. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L10c | Usage: CALL ACORR (Y, N, NC, SCRAT, NS) | On-line doc: CALL GAMSDOC ACORR (or @PRT STATLIB*DOC.ACORR) | Tests: STATLIB*TEST.DEMO4 | Access: LIB NBS*STATLIB
- ACORRD** Performs autocorrelation analysis of a series differenced with a user-controlled differencing operation. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L10c | Usage: CALL ACORRD (Y, N, NC, NDIFAC, ND, IOD, SCRAT, NS) | On-line doc: CALL GAMSDOC ACORRD (or @PRT STATLIB*DOC.ACORRD) | Tests: STATLIB*TEST.DEMO4 | Access: LIB NBS*STATLIB
- ACORRS** Performs autocorrelation analysis of a series, with computed results returned to the user. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L10c | Usage: CALL ACORRS (Y, N, NC, RHO, SCRAT, NS) | On-line doc: CALL GAMSDOC ACORRS (or @PRT STATLIB*DOC.ACORRS) | Tests: STATLIB*TEST.DEMO4 | Access: LIB NBS*STATLIB
- ACOSH** Hyperbolic cosine. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DACOSH. | Class(es): C4c | Usage: R=ACOSH(X) | On-line doc: CALL GAMSDOC ACOSH (or @PRT

CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

- ACOSH** Computes hyperbolic arccosine, arccosh. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DACOSH. | Class(es): C4c | Usage: $X = \text{ACOSH}(X)$ | On-line doc: CALL GAMSDOC ACOSH (or @PRT PORT*DOC.ACOSH) | Access: LIB NBS*PORT
- ACRDAN** Analysis of one-way classification design data. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a1a | Usage: CALL ACRDAN (Y,NT,N,TM,WTV,S,GM,NDF,IER) | On-line doc: CALL GAMSDOC ACRDAN (or @PRT IMSL*DOC.ACRDAN) | Access: LIB NBS*IMSL
- ACSPEC** Computes the series autospectrum from the Fourier transform of the user-supplied autocorrelation function, with user-supplied lag window truncation values. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L10f | Usage: CALL ACSPEC (RHO, NC, NW, LAGS, Y, N, SCRAT, NS) | On-line doc: CALL GAMSDOC ACSPEC (or @PRT STATLIB*DOC.ACSPEC) | Tests: STATLIB*TEST.DEMO4 | Access: LIB NBS*STATLIB
- ACTRST** Contrast estimates and sums of squares. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a1a1 | Usage: CALL ACTRST (T,NR,N,IP,P,IP,Q,SQ) | On-line doc: CALL GAMSDOC ACTRST (or @PRT IMSL*DOC.ACTRST) | Access: LIB NBS*IMSL
- AFACN** Full factorial plan analysis. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a2a1b | Usage: CALL AFACN (IOPT,NF,NL,Y,SS,NDF,IER) | On-line doc: CALL GAMSDOC AFACN (or @PRT IMSL*DOC.AFACN) | Access: LIB NBS*IMSL
- AFACT** Full factorial plan analysis - easy to use version. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a2a1b | Usage: CALL AFACT (NF,NL,IJOB,Y,GMEAN,YMEANS,INDEX,STAT,IS,IER) | On-line doc: CALL GAMSDOC AFACT (or @PRT IMSL*DOC.AFACT) | Access: LIB NBS*IMSL
- AGBACP** Analysis of balanced complete experimental design structure data. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a2a1 | Usage: CALL AGBACP (NF,NL,IA,Y,IWL,LST,LOC,SS,NDF,IER) | On-line doc: CALL GAMSDOC AGBACP (or @PRT IMSL*DOC.AGBACP) | Access: LIB NBS*IMSL
- AGLMOD** General linear model analysis. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a4a | Usage: CALL AGLMOD (X,IX,NV,Y,XL,IL,BETA,SS,VARB,WK,IPR,IER) | On-line doc: CALL GAMSDOC AGLMOD (or @PRT IMSL*DOC.AGLMOD) | Access: LIB NBS*IMSL
- AGVACL** One or two-sided interval estimate of a variance component. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a1a2 | Usage: CALL AGVACL (V,FDF,S,CF,IOP,STAT,IER) | On-line doc: CALL GAMSDOC AGVACL (or @PRT IMSL*DOC.AGVACL) | Access: LIB NBS*IMSL
- AGXPM** Expected mean squares for balanced complete design models. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a2a1 | Usage: CALL AGXPM (IOPT,NF,M,IA,NL,INL,CMS,IORD,IEMS,STAT,IS,ERTM,IE,IER) | On-line doc: CALL GAMSDOC AGXPM (or @PRT IMSL*DOC.AGXPM) | Access: LIB NBS*IMSL
- AI** Airy function. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DAL. | Class(es): C10d | Usage: $R = \text{AI}(X)$ | On-line doc: CALL GAMSDOC AI (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- AIE** Exponentially scaled Airy function. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DAIE. | Class(es): C10d | Usage: $R = \text{AIE}(X)$ | On-line doc: CALL GAMSDOC AIE (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- ALBETA** Log Beta. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DLBETA. | Class(es): C7b | Usage: $R = \text{ALBETA}(A,B)$ | On-line doc: CALL GAMSDOC ALBETA (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- ALGAMA** Evaluate the log (base e) of the absolute value of the gamma function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): C7a | Usage: $X = \text{ALGAMA}(Y)$ | On-line doc: CALL GAMSDOC ALGAMA (or @PRT IMSL*DOC.ALGAMA) | Access: LIB NBS*IMSL
- ALGAMS** $G = \ln \text{abs}(\text{Gamma}(x))$, Log abs gamma with sign of gamma $S = \text{sign Gamma}(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DLGAMS. | Class(es): C7a | Usage: CALL ALGAMS (X,G,S) | On-line doc: CALL GAMSDOC ALGAMS (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- ALI** Integral over the range 0 to X of $(1/\ln t)$ dt. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DLI. | Class(es): C5 | Usage: $R = \text{ALI}(X)$ | On-line doc: CALL GAMSDOC ALI (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- ALNGAM** Ln absolute value of $\text{Gamma}(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DLNGAM. | Class(es): C7a | Usage: $R = \text{ALNGAM}(X)$ | On-line doc: CALL GAMSDOC ALNGAM (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- ALNREL** $\ln(1 + x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DLNREL. | Class(es): C4b | Usage: $R = \text{ALNREL}(X)$ | On-line doc: CALL GAMSDOC ALNREL (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- ALOG** $\ln(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4b | Usage: $R = \text{ALOG}(X)$ | On-line doc: CALL GAMSDOC ALOG (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

- ALOG10** Common logarithm Log to the base 10 of x. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4b | Usage: R=ALOG10(X) | On-line doc: CALL GAMSDOC ALOG10 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- ALSQAN** Analysis of Latin square design data. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a2b1a | Usage: CALL ALSQAN (Y,NR,NT,IND,EM,GM,S,NDF,IER) | On-line doc: CALL GAMSDOC ALSQAN (or @PRT IMSL*DOC.ALSQAN) | Access: LIB NBS*IMSL
- AMEANS** Preparation of a set of unbalanced data for analysis by the method of unweighted means. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a4a | Usage: CALL AMEANS (Y,N,K,YM,HN,SS,IER) | On-line doc: CALL GAMSDOC AMEANS (or @PRT IMSL*DOC.AMEANS) | Access: LIB NBS*IMSL
- ANCOV1** Covariance analysis for one-way classification design data. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a3 | Usage: CALL ANCOV1 (XY,NOP,IXY,XYM,IXYM,TM,SXY,VARB,VART,SS,NDF,WK, IER) | On-line doc: CALL GAMSDOC ANCOV1 (or @PRT IMSL*DOC.ANCOV1) | Access: LIB NBS*IMSL
- ANESTE** Analysis of completely nested design data with equal numbers in the subclasses. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a2a1c | Usage: CALL ANESTE(NF,NL,Y,S,NDR,IER) | On-line doc: CALL GAMSDOC ANESTE (or @PRT IMSL*DOC.ANESTE) | Access: LIB NBS*IMSL
- ANESTU** Analysis of completely nested design data with unequal number in the subclasses. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a2a1c | Usage: CALL ANESTU(NF,NL,Y,GM,S,NDF,EMS,IWK,IER) | On-line doc: CALL GAMSDOC ANESTU (or @PRT IMSL*DOC.ANESTU) | Access: LIB NBS*IMSL
- AORDR** Reordering of the data obtained from any balanced complete experimental design. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a2a1 | Usage: CALL AORDR (NF,NL,IORD,YIN,JORD,YOUT,IWK,IER) | On-line doc: CALL GAMSDOC AORDR (or @PRT IMSL*DOC.AORDR) | Access: LIB NBS*IMSL
- AOVONEWAY** Performs a one-way analysis of variance and prints standard results. | Command in MINITAB Proprietary interactive system. | Class(es): L7a1a | Usage: AOVOneway on data in C,...,C | On-line doc: HELP AOVONEWAY (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- ARCBAN** Analysis of two-way classification design data. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a2a1a | Usage: CALL ARCBAN (Y,NR,NB,NT,EM,GM,S,NDF,IER) | On-line doc: CALL GAMSDOC ARCBAN (or @PRT IMSL*DOC.ARCBAN) | Access: LIB NBS*IMSL
- ARCOS** Computes arccos(x), answer in radians. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DARCOS. | Class(es): C4a | Usage: X = ARCOS (X) | On-line doc: CALL GAMSDOC ARCOS (or @PRT PORT*DOC.ARCOS) | Access: LIB NBS*PORT
- ARIMA** Fits non-seasonal and seasonal models to a time series with p the order of the AR part, d the number of differences, q the order of the MA part and with optional seasonality with period S, AR order P, number of differences D, and MA order Q. Options: starting values, forecasting, save results. | Command in MINITAB Proprietary interactive system. | Class(es): L10e | Usage: ARIMA P=K, D=K, Q=K [P=K, D=K, Q=K, S=K] for data in C [put residuals in C [put predicted values in C [put estimated parameters in C]]] ; subcommands CONSTANT or NOCONSTANT; STARTING values in C; FORECAST [forecast origin = K] up to K leads ahead [store forecasts in C [confidence limits in C,C]]. | On-line doc: HELP ARIMA (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- ARIMAE** Performs least squares estimation of the parameters in an ARIMA (Box-Jenkins) model using an adaptation of Pack's code. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L10e1 | Usage: CALL ARIMAE (Y, N, NAR, NDIFAC, NMA, ND, IOD, INC, NP, IOPA, COEF, RES, NRES, STOPSS, STOPCR, MIT, SCRAT, NS) | On-line doc: CALL GAMSDOC ARIMAE (or @PRT STATLIB*DOC.ARIMAE) | Tests: STATLIB*TEST.DEMO3 | Access: LIB NBS*STATLIB
- ARIMAF** Performs minimum mean square error forecasts for a given (fitted) ARIMA (Box-Jenkins) model, using an adaptation of Pack's code. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L10e2 | Usage: CALL ARIMAF (Y, N, NAR, NDIFAC, NMA, ND, IOD, INC, NP, IOPA, COEF, ICI, NF, NTO, ITO, FOR, FORLCL, FORUCL, IRDIM, ICDIM, NU, YN, SCRAT, NS) | On-line doc: CALL GAMSDOC ARIMAF (or @PRT STATLIB*DOC.ARIMAF) | Tests: STATLIB*TEST.DEMO3 | Access: LIB NBS*STATLIB
- ARSIN** Computes arcsin(x), answer in radians. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DARSIN. | Class(es): C4a | Usage: X = ARSIN (X) | On-line doc: CALL GAMSDOC ARSIN (or @PRT PORT*DOC.ARSIN) | Access: LIB NBS*PORT
- ASINH** Hyperbolic sine. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DASINH. | Class(es): C4c | Usage: R=ASINH(X) | On-line doc: CALL GAMSDOC ASINH (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- ASINH** Computes hyperbolic arcsine, acrsin(x). | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DASINH. | Class(es): C4c | Usage: X = ASINH (X) | On-line doc: CALL GAMSDOC ASINH (or @PRT PORT*DOC.ASINH) | Access: LIB NBS*PORT
- ASNKMC** Student-Newman-Keuls multiple comparison test. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a1a1 | Usage: CALL ASNKMC (Y,M,SY,NDF,ALPHA,IR,IW,IER) | On-line doc: CALL GAMSDOC ASNKMC (or @PRT IMSL*DOC.ASNKMC) | Access: LIB NBS*IMSL
- ASPEC** Computes the series autospectrum from the Fourier transform of the autocorrelation function, using the Jenkins and Watts window closing technique. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L10f | Usage: CALL ASPEC (Y, N) | On-line doc: CALL GAMSDOC ASPEC (or @PRT STATLIB*DOC.ASPEC) | Tests: STATLIB*TEST.DEMO4 | Access: LIB NBS*STATLIB

- ASPECS** Computes the series autospectrum from the Fourier transform of the autocorrelation function, with user-supplied lag window truncation values. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L10f | Usage: CALL ASPECS (Y, N, NW, LAGS, SCRAT, NS) | On-line doc: CALL GAMSDOC ASPECS (or @PRT STATLIB*DOC.ASPECS) | Tests: STATLIB*TEST.DEMO4 | Access: LIB NBS*STATLIB
- ATANH** Arc hyperbolic tangent. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DATANH. | Class(es): C4c | Usage: R=ATANH(X) | On-line doc: CALL GAMSDOC ATANH (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- ATANH** Computes hyperbolic arctangent, $\text{acrtanh}(x)$. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DATANH. | Class(es): C4c | Usage: X = ATANH (X) | On-line doc: CALL GAMSDOC ATANH (or @PRT PORT*DOC.ATANH) | Access: LIB NBS*PORT
- AUTOCO** Computes the sample autocorrelation coefficient of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L10c | Usage: CALL AUTOCO(X,N,IWRITE,XAUTOC) | On-line doc: CALL GAMSDOC AUTOCO (or @PRT DATAPAC*DOC.AUTOCO) | Access: LIB NBS*DAPAC

B

- B2INK** Computes parameters of a piecewise-polynomial that interpolates a given set of two-dimensional gridded data. (Use B2VAL to evaluate function.). | Portable single precision Fortran subprogram in TENSORBS sublibrary of CMLIB library. Double precision version is DB2INK. | Class(es): E2a | Usage: CALL B2INK(X,NX,Y,NY,FCN,LDF,KX,KY,TX,TY,BCOEF,WORK,IFLAG) | On-line doc: CALL GAMSDOC B2INK (or @PRT CMLIB*DOC.B2INK/TENSORBS) | Tests: CMLIB*TEST-SOURCE.\$Q1/TENSORBS, CMLIB*TEST-SOURCE.\$Q2/TENSORBS | Access: LIB NBS*CMLIB | See also: B2VAL
- B2VAL** Evaluates the two-dimensional interpolating function computed by B2INK or one of its partial derivatives. | Portable single precision Fortran subprogram in TENSORBS sublibrary of CMLIB library. Double precision version is DB2VAL. | Class(es): E3 | Usage: CALL B2VAL(XVAL,YVAL,IDX,IDY,TX,TY,NX,NY,KX,KY,BCOEF,WORK) | On-line doc: CALL GAMSDOC B2VAL (or @PRT CMLIB*DOC.B2VAL/TENSORBS) | Tests: CMLIB*TEST-SOURCE.\$Q1/TENSORBS, CMLIB*TEST-SOURCE.\$Q2/TENSORBS | Access: LIB NBS*CMLIB
- B3INK** Computes parameters of a piecewise-polynomial that interpolates a given set of three-dimensional gridded data. (Use B3VAL to evaluate function.). | Portable single precision Fortran subprogram in TENSORBS sublibrary of CMLIB library. Double precision version is DB3INK. | Class(es): E2a | Usage: CALL B3INK(X,NX,Y,NY,FCN,LDF,KX,KY,TX,TY,BCOEF,WORK,IFLAG) | On-line doc: CALL GAMSDOC B3INK (or @PRT CMLIB*DOC.B3INK/TENSORBS) | Tests: CMLIB*TEST-SOURCE.\$Q1/TENSORBS, CMLIB*TEST-SOURCE.\$Q2/TENSORBS | Access: LIB NBS*CMLIB | See also: B3VAL
- B3VAL** Evaluates the three-dimensional interpolating function computed by B3INK or one of its partial derivatives. | Portable single precision Fortran subprogram in TENSORBS sublibrary of CMLIB library. Double precision version is DB3VAL. | Class(es): E3 | Usage: CALL B3VAL(XVAL,YVAL,IDX,IDY,TX,TY,NX,NY,KX,KY,BCOEF,WORK) | On-line doc: CALL GAMSDOC B3VAL (or @PRT CMLIB*DOC.B3VAL/TENSORBS) | Tests: CMLIB*TEST-SOURCE.\$Q1/TENSORBS, CMLIB*TEST-SOURCE.\$Q2/TENSORBS | Access: LIB NBS*CMLIB
- BAKVEC** Forms eigenvectors of certain real non-symmetric tridiagonal matrix from symmetric tridiagonal matrix output from FIG1. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL BAKVEC(NM,N,T,E,M,Z,IERR) | On-line doc: CALL GAMSDOC BAKVEC (or @PRT CMLIB*DOC.BAKVEC/EISPACK) | Access: LIB NBS*CMLIB | See also: FIG1
- BALANC** Balances a general real matrix and isolates eigenvalues whenever possible. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1a | Usage: CALL BALANC(NM,N,A,LOW,IGH,SCALE) | On-line doc: CALL GAMSDOC BALANC (or @PRT CMLIB*DOC.BALANC/EISPACK) | Access: LIB NBS*CMLIB
- BALBAK** Forms eigenvectors of real general matrix from eigenvectors of matrix output from BALANC. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL BALBAK(NM,N,LOW,IGH,SCALE,M,Z) | On-line doc: CALL GAMSDOC BALBAK (or @PRT CMLIB*DOC.BALBAK/EISPACK) | Access: LIB NBS*CMLIB | See also: BALANC
- BANDR** Reduces real symmetric band matrix to symmetric tridiagonal matrix and, optionally, accumulates orthogonal similarity transformations. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1b1 | Usage: CALL BANDR(NM,N,MB,A,D,E,E2,MATZ,Z) | On-line doc: CALL GAMSDOC BANDR (or @PRT CMLIB*DOC.BANDR/EISPACK) | Access: LIB NBS*CMLIB
- BANDV** Forms eigenvectors of real symmetric band matrix associated with a set of ordered approximate eigenvalues by inverse iteration. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c3 | Usage: CALL BANDV(NM,N,MBW,A,E21,M,W,Z,IERR,NV,RV,RV6) | On-line doc: CALL GAMSDOC BANDV (or @PRT CMLIB*DOC.BANDV/EISPACK) | Access: LIB NBS*CMLIB
- BDCOU1** Tally of observations into a one-way frequency table. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2b | Usage: CALL BDCOU1 (X,N,K,DIV,BU,BL,TAB,IER) | On-line doc: CALL GAMSDOC BDCOU1 (or @PRT IMSL*DOC.BDCOU1) | Access: LIB NBS*IMSL
- BDCOU2** Tally of observations into a two-way frequency table. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2b | Usage: CALL BDCOU2 (X,Y,N,K1,K2,DIVX,DIVY,XU,XL,YU,YL,IT,TAB,IER) | On-line doc: CALL GAMSDOC BDCOU2 (or @PRT IMSL*DOC.BDCOU2) | Access: LIB NBS*IMSL
- BDLTV** Produce letter value summary. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1a1 | Usage: CALL BDLTV (X,N,NUM,SUMRY,IER) | On-line doc: CALL GAMSDOC BDLTV (or @PRT IMSL*DOC.BDLTV) | Access: LIB NBS*IMSL
- BDTAB** Computations of frequencies of multivariate data. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2b | Usage: CALL BDTAB (X, M, KMAX, NOPT, ICNT, K, ITAB, VECVAL, IVEC, VARVAL, IVAR, IDIST, WK, IER) | On-line doc: CALL GAMSDOC BDTAB (or @PRT IMSL*DOC.BDTAB) | Access: LIB NBS*IMSL
- BDTRGI** Transgeneration of the columns of a matrix (in-core version). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2a | Usage: CALL BDTRGI (X,N,IX,NT,ITRG,IT,C,INFER,IER) | On-line doc: CALL GAMSDOC BDTRGI (or @PRT IMSL*DOC.BDTRGI) | Access: LIB NBS*IMSL
- BDTRGO** Transgeneration of the columns of a matrix (out-of-core version). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2a | Usage: CALL BDTRGO (X,NT,ITRG,IT,C,IER) | On-line doc: CALL GAMSDOC BDTRGO (or @PRT IMSL*DOC.BDTRGO) | Access: LIB NBS*IMSL
- BDTWT** Computations of a two-way frequency table. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2b | Usage: CALL BDTWT (N1, N2, K, ITAB, VECVAL, IVEC, VARVAL, IVAR, MATFRQ, IM, IRTOT, ICTOT, IALTOT, CHISQ,

- P, IER) | On-line doc: CALL GAMSDOC BDTWT (or @PRT IMSL*DOC.BDTWT) | Access: LIB NBS*IMSL
- BECOR** Estimates of means, standard deviations, and correlation coefficients (out-of-core version). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1e1 | Usage: CALL BECOR (X,N,M,I,IND,TEMP,XMD,SD,RD,IER) | On-line doc: CALL GAMSDOC BECOR (or @PRT IMSL*DOC.BECOR) | Access: LIB NBS*IMSL
- BECORI** Estimates of means, standard deviations, and correlation coefficients (in-core version). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1e1 | Usage: CALL BECORI (X,N,M,IX,XM,S,R,IER) | On-line doc: CALL GAMSDOC BECORI (or @PRT IMSL*DOC.BECORI) | Access: LIB NBS*IMSL
- BECOV** Means and variance-covariance matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1e1 | Usage: CALL BECOV (X,IX,NBR,TEMP,XM,VCV,IER) | On-line doc: CALL GAMSDOC BECOV (or @PRT IMSL*DOC.BECOV) | Access: LIB NBS*IMSL
- BECOVW** Means and variance-covariance or correlation matrix from data possibly containing missing observations, with weighting on option. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1e2 | Usage: CALL BECOVW (X, IX, WT, NBR, XMISS, XM, VCV, INCD, WK, IER) | On-line doc: CALL GAMSDOC BECOVW (or @PRT IMSL*DOC.BECOVW) | Access: LIB NBS*IMSL
- BECTR** Tetrachoric correlation coefficient estimation. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4b1a14 | Usage: CALL BECTR (N,U,V,HU,HV,IOPT,R,RS,K,IER) | On-line doc: CALL GAMSDOC BECTR (or @PRT IMSL*DOC.BECTR) | Access: LIB NBS*IMSL
- BECVL** Variances and covariances of linear functions (out-of-core version). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1e1 | Usage: CALL BECVL (X,M,Y,C,R,V) | On-line doc: CALL GAMSDOC BECVL (or @PRT IMSL*DOC.BECVL) | Access: LIB NBS*IMSL
- BECVLI** Variances and covariances of linear functions (in-core version). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1e1 | Usage: CALL BECVLI (X,N,M,IX,C,R,IOPT,V) | On-line doc: CALL GAMSDOC BECVLI (or @PRT IMSL*DOC.BECVLI) | Access: LIB NBS*IMSL
- BEGRPS** Moments estimation for grouped data with and without Sheppards corrections. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1a3 | Usage: CALL BEGRPS (N,C,CI,U,UC,IER) | On-line doc: CALL GAMSDOC BEGRPS (or @PRT IMSL*DOC.BEGRPS) | Access: LIB NBS*IMSL
- BEIGRP** Estimation of basic statistical parameters using grouped data. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1a3 | Usage: CALL BEIGRP (F,Y,K,YLM,WID,IOPT,STAT,IER) | On-line doc: CALL GAMSDOC BEIGRP (or @PRT IMSL*DOC.BEIGRP) | Access: LIB NBS*IMSL
- BEIUGR** Estimation of basic statistical parameters using ungrouped data. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1a1 | Usage: CALL BEIUGR (Y,N,IOPT,STAT,IER) | On-line doc: CALL GAMSDOC BEIUGR (or @PRT IMSL*DOC.BEIUGR) | Access: LIB NBS*IMSL
- BELBIN** Interval estimate of the parameter p of the binomial distribution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1a2 | Usage: CALL BELBIN (NTRIAL,NX,ALPHA,PHAT,PLOWER,PUPPER,IER) | On-line doc: CALL GAMSDOC BELBIN (or @PRT IMSL*DOC.BELBIN) | Access: LIB NBS*IMSL
- BELPOS** Interval estimate of the parameter lambda of the Poisson distribution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1a16 | Usage: CALL BELPOS (NPOIS,NN,ALPHA,RLAMHT,RLAMLR,RLAMUP,IER) | On-line doc: CALL GAMSDOC BELPOS (or @PRT IMSL*DOC.BELPOS) | Access: LIB NBS*IMSL
- BEMDP** Median polish of a two-way table. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L0d | Usage: CALL BEMDP (TAB,IR,IC,IRTAB,MAXIT,WK,IER) | On-line doc: CALL GAMSDOC BEMDP (or @PRT IMSL*DOC.BEMDP) | Access: LIB NBS*IMSL
- BEMIRI** Estimates means, simple regression coefficients, their intercepts, standard errors of the regression coefficients, and standard deviations for arrays which contain missing values (in-core version). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a1a1b | Usage: CALL BEMIRI (X,N,M,IX,XMEAN,B,A,S,IBAS,INCD,IER) | On-line doc: CALL GAMSDOC BEMIRI (or @PRT IMSL*DOC.BEMIRI) | Access: LIB NBS*IMSL
- BEMIRO** Estimates means, simple regression coefficients, their intercepts, standard errors of the regression coefficients, and standard deviations for arrays which contain missing values (out-of-core version). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a1a1b | Usage: CALL BEMIRO (X,N,M,IND,XMEAN,B,A,S,IBAS,INCD,IER) | On-line doc: CALL GAMSDOC BEMIRO (or @PRT IMSL*DOC.BEMIRO) | Access: LIB NBS*IMSL
- BEMMI** Estimates of means, std. devs., correlation coefficients, and coefficients of skewness and kurtosis from a data matrix containing missing observations (in-core version). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1e2 | Usage: CALL BEMMI (X,N,M,IX,XMEAN,U,A,INCD,IER) | On-line doc: CALL GAMSDOC BEMMI (or @PRT IMSL*DOC.BEMMI) | Access: LIB NBS*IMSL
- BEMMO** Estimates of means, std. devs., correlation coefficients, and coefficients of skewness and kurtosis from a data matrix containing missing observations (out-of-core version). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1e2 | Usage: CALL BEMMO (X,N,M,IND,XMEAN,U,A,INCD,WK,IER) | On-line doc: CALL GAMSDOC BEMMO (or @PRT IMSL*DOC.BEMMO) | Access: LIB NBS*IMSL
- BEMNON** Location (mean) inferences using a sample from a normal population with known variance. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1a14 | Usage: CALL BEMNON (Y,N,IOP,CRIT,M,YMN,STAT,IER) | On-line doc: CALL

GAMSDOC BEMNON (or @PRT IMSL*DOC.BEMNON) | Access: LIB NBS*IMSL

BEMSON Mean and variance inferences using a sample from a normal population. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1a14 | Usage: CALL BEMSON (Y,N,IOP,CRIT,M,PAR,STAT,NDF,IER) | On-line doc: CALL GAMSDOC BEMSON (or @PRT IMSL*DOC.BEMSON) | Access: LIB NBS*IMSL

BENSON Variance inferences using a sample from a normal population with known mean. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1a14 | Usage: CALL BENSON (Y,N,IOP,CRIT,M,VAR,STAT,NDF,IER) | On-line doc: CALL GAMSDOC BENSON (or @PRT IMSL*DOC.BENSON) | Access: LIB NBS*IMSL

BEPAT Mean and variance inferences using samples from each of two normal populations with unequal variances. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4b1a14 | Usage: CALL BEPAT (Y,N,IOP,CRIT,STAT,IER) | On-line doc: CALL GAMSDOC BEPAT (or @PRT IMSL*DOC.BEPAT) | Access: LIB NBS*IMSL

BEPET Mean and variance inferences using samples from each of two normal populations with equal variances. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4b1a14 | Usage: CALL BEPET (Y,N,IOP,CRIT,STAT,NDF,IER) | On-line doc: CALL GAMSDOC BEPET (or @PRT IMSL*DOC.BEPET) | Access: LIB NBS*IMSL

BESCI Modified Bessel functions, I, of complex argument and integer order. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DBESCI. | Class(es): C10b2 | Usage: CALL BESCI (XR, XI, NB, BR, BI) | On-line doc: CALL GAMSDOC BESCI (or @PRT PORT*DOC.BESCI) | Access: LIB NBS*PORT

BESCJ Bessel functions, J, of complex argument and integer order. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DBESCJ. | Class(es): C10a2 | Usage: CALL BESCJ (XR, XI, NB, BR, BI) | On-line doc: CALL GAMSDOC BESCJ (or @PRT PORT*DOC.BESCJ) | Access: LIB NBS*PORT

BESIO Hyperbolic Bessel Function of first kind, order zero $I_0(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESIO. | Class(es): C10b1 | Usage: R=BESIO(X) | On-line doc: CALL GAMSDOC BESIO (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

BESIOE Modified hyperbolic Bessel functions of special integer order scaled by an exponential: $e^{-abs(x)} * I_0(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBSIOE. | Class(es): C10b1 | Usage: R = BESIOE(X) | On-line doc: CALL GAMSDOC BESIOE (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

BESI1 Modified (hyperbolic) Bessel Functions of special integer order $I_1(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESI1. | Class(es): C10b1 | Usage: R = BESI1(X) | On-line doc: CALL GAMSDOC BESI1 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

BESI1E Bessel function scaled by an exponential $e^{-abs(x)} * I_1(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBSI1E. | Class(es): C10b1 | Usage: R = BESI1E(X) | On-line doc: CALL GAMSDOC BESI1E (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

BESJ Computes an N member sequence of J Bessel fcn's, $J_{sub}(\alpha+k-1)/x$ $k=1,...,N$ for $x, \alpha \geq 0$. Uses internal double precision arith. | Portable single precision Fortran subprogram in AMOSLIB sublibrary of CMLIB library. | Class(es): C10a3 | Usage: CALL BESJ(X,ALPHA,N,Y,NZ) | On-line doc: CALL GAMSDOC BESJ (or @PRT CMLIB*DOC.BESJ/AMOSLIB) | Tests: CMLIB*TEST-SOURCE.BESJ/AMOSLIB | Access: LIB NBS*CMLIB

BESJ0 Bessel function of special integer order: First kind, order zero $J_0(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESJ0. | Class(es): C10a1 | Usage: R = BESJ0(X) | On-line doc: CALL GAMSDOC BESJ0 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

BESJ1 Bessel function of special integer order; first kind, order 1 $J_1(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESJ1. | Class(es): C10a1 | Usage: R = BESJ1(X) | On-line doc: CALL GAMSDOC BESJ1 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

BESK0 Modified (hyperbolic) Bessel function of special integer order Third kind, order 0. $K_0(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESK0. | Class(es): C10b1 | Usage: R = BESK0(X) | On-line doc: CALL GAMSDOC BESK0 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

BESK0E Modified (hyperbolic) Bessel function of special integer order scaled by an exponential. Third kind order 0. $e^{**x} * K_0(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBSK0E. | Class(es): C10b1 | Usage: R = BESK0E(X) | On-line doc: CALL GAMSDOC BESK0E (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

BESK1 Modified (hyperbolic) Bessel function of special integer order; third kind, order one $K_1(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESK1. | Class(es): C10b1 | Usage: R = BESK1(X) | On-line doc: CALL GAMSDOC BESK1 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

BESK1E Modified (hyperbolic) Bessel function of special integer order scaled by an exponential. Third kind, order one. $e^{**x} * K_1(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBSK1E. | Class(es): C10b1 | Usage: R = BESK1E(X) | On-line doc: CALL GAMSDOC BESK1E (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

BESKES Seq of Bessel function scaled by exponential. Abs(n) values are computed for $n > 0, i = 0, 1...n-1$ for $n < 0, i = 0, -1...n+1$ $e^{**x} * K_{sub} v+1(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version

- is DBSKES. | Class(es): C10b3 | Usage: CALL BESKES(XNU, X, N, BK) | On-line doc: CALL GAMSDOC BESKES (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- BESKS** Sequences of Bessel Functions. Abs(n) values are computed. For $n > 0$, $i = 0, 1, \dots, n-1$ for $n < 0$, $i = 0, -1, \dots, -n+1$ K sub $v+1$ (x). | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESKS. | Class(es): C10b3 | Usage: CALL BESKS(XNU,X,N,BK) | On-line doc: CALL GAMSDOC BESKS (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- BESRB** Biserial and point-biserial correlation coefficients for a qualitatively dichotomized variable and a numerically measurable and classified variable. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4d | Usage: CALL BESRB (N,A,IA,STAT,IER) | On-line doc: CALL GAMSDOC BESRB (or @PRT IMSL*DOC.BESRB) | Access: LIB NBS*IMSL
- BESRI** Modified Bessel functions, I, of real argument and integer order. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DBESRI. | Class(es): C10b1 | Usage: CALL BESRI (X, NB, B) | On-line doc: CALL GAMSDOC BESRI (or @PRT PORT*DOC.BESRI) | Access: LIB NBS*PORT
- BESRJ** Bessel functions, J, of real argument and integer order. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DBESRJ. | Class(es): C10a1 | Usage: CALL BESRJ (X, NB, B) | On-line doc: CALL GAMSDOC BESRJ (or @PRT PORT*DOC.BESRJ) | Access: LIB NBS*PORT
- BESRN** Biserial correlation coefficient for a qualitatively dichotomized variable and a numerically or qualitatively classified variable. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4d | Usage: CALL BESRN (N,A,IA,STAT,IER) | On-line doc: CALL GAMSDOC BESRN (or @PRT IMSL*DOC.BESRN) | Access: LIB NBS*IMSL
- BESTA2** Computations of confidence intervals and other basic statistics using output from IMSL routine BESTAT. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a2 | Usage: CALL BESTA2 (NVAR, IOPT, CLM, CLV, STATS, IS, RNG, CV, CIM, CIV, IER) | On-line doc: CALL GAMSDOC BESTA2 (or @PRT IMSL*DOC.BESTA2) | Access: LIB NBS*IMSL | See also: BESTAT
- BESTAT** Computations of basic univariate statistics from data possibly containing missing values, with weighting on option. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1a2 | Usage: CALL BESTAT (X, IX, WT, NBR, XMISS, STATS, IS, WK, IER) | On-line doc: CALL GAMSDOC BESTAT (or @PRT IMSL*DOC.BESTAT) | Access: LIB NBS*IMSL
- BESY0** Modified (hyperbolic) Bessel function of special integer order. Second kind, order 0. $Y_{\text{sub } 0}(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESY0. | Class(es): C10a1 | Usage: R = BESY0(X) | On-line doc: CALL GAMSDOC BESY0 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- BESY1** Modified (hyperbolic) Bessel function of special integer order. Second kind, order 1. $Y_{\text{sub } 1}(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESY1. | Class(es): C10a1 | Usage: R = BESY1(X) | On-line doc: CALL GAMSDOC BESY1 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- BETA** $\text{Beta}(a,b) = (\text{Gamma}(a) * \text{Gamma}(b)) / \text{Gamma}(a + b)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBETA. | Class(es): C7b | Usage: R = BETA(A,B) | On-line doc: CALL GAMSDOC BETA (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- BETAI** Incomplete Beta. $I_{\text{sub } x}(a,b) = B_{\text{sub } x}(a,b) / B(a,b)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBETAI. | Class(es): C7f | Usage: R = BETAI(X,A,B) | On-line doc: CALL GAMSDOC BETAI (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- BETRAN** Generates a random sample of size N from the beta distribution with parameters ALPHA and BETA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a2 | Usage: CALL BETRAN(N,ALPHA,BETA,ISTART,X) | On-line doc: CALL GAMSDOC BETRAN (or @PRT DATAPAC*DOC.BETRAN) | Access: LIB NBS*DAPAC
- BFQAD** Integrates function times derivative of B-spline from X1 to X2. The B-spline is in "B" representation. | Portable single precision Fortran subprogram in BSPLINE sublibrary of CMLIB library. Double precision version is DBFQAD. | Class(es): H2a2a1 E3 K6 | Usage: CALL BFQAD(F,T,BCOEF,N,K,ID,X1,X2,TOL,QUAD,IERR,WORK) | On-line doc: CALL GAMSDOC BFQAD (or @PRT CMLIB*DOC.BFQAD/BSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/BSPLINE | Access: LIB NBS*CMLIB
- BI** Bairy function $Bi(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBI. | Class(es): C10d | Usage: R = BI(X) | On-line doc: CALL GAMSDOC BI (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- BIE** Exponentially scaled Airy function. Bairy $Bi(x)$, $x \leq 0$; $\exp(-2/3 * x^{3/2}) * Bi(x)$, $x \geq 0$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBIE. | Class(es): C10d | Usage: R = BIE(X) | On-line doc: CALL GAMSDOC BIE (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- BINCDF** Computes the cumulative distribution function value at X for the binomial distribution with parameters P and N. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1b | Usage: CALL BINCDF(X,P,N,CDF) | On-line doc: CALL GAMSDOC BINCDF (or @PRT DATAPAC*DOC.BINCDF) | Access: LIB NBS*DAPAC
- BINOM** Binomial $n! / (m! * (n-m)!)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBINOM. | Class(es): C1 | Usage: R = BINOM(N, M) | On-line doc: CALL GAMSDOC BINOM (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- BINOMIAL** Prints table of binomial probabilities and cumulative distribution function, and optionally saves results. | Command in MINITAB

Proprietary interactive system. Class(es): L6a1b | Usage: BINomial probabilities for $N=K$, $P=K$ [store probabilities in C] | On-line doc: HELP BINOMIAL (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

BINPPF Computes the percent point function value at P for the binomial distribution with parameters PPAR and N. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a2b | Usage: CALL BINPPF(P,PPAR,N,PPF) | On-line doc: CALL GAMSDOC BINPPF (or @PRT DATAPAC*DOC.BINPPF) | Access: LIB NBS*DAPAC

BINRAN Generates a random sample of size N from the binomial distribution with parameters P and NPAR. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a2 | Usage: CALL BINRAN(N,P,NPAR,ISTART,X) | On-line doc: CALL GAMSDOC BINRAN (or @PRT DATAPAC*DOC.BINRAN) | Access: LIB NBS*DAPAC

BINT4 Computes B-spline which interpolates given X,Y data with various end conditions. The "B" representation is used. | Portable single precision Fortran subprogram in BSPLINE sublibrary of CMLIB library. Double precision version is DBINT4. | Class(es): E1a | Usage: CALL BINT4(X,Y,NDATA,IBCL,IBCR,FBCF,FBCR,KNTOPT,T,BCOEF,N,K,W) | On-line doc: CALL GAMSDOC BINT4 (or @PRT CMLIB*DOC.BINT4/BSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/BSPLINE | Access: LIB NBS*CMLIB | See also: BVALU for evaluation. See package documentation for other facilities.

BINTK Produces B-spline coefficients of k-th order B-spline with given knots and with values at given points. | Portable single precision Fortran subprogram in BSPLINE sublibrary of CMLIB library. Double precision version is DBINTK. | Class(es): E1a | Usage: CALL BINTK(X,Y,T,N,K,BCOEF,Q,WORK) | On-line doc: CALL GAMSDOC BINTK (or @PRT CMLIB*DOC.BINTK/BSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/BSPLINE | Access: LIB NBS*CMLIB | See also: BVALU for evaluation. See package documentation for other facilities.

BISECT Compute eigenvalues of symmetric tridiagonal matrix in given interval using Sturm sequencing. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 D4c2a | Usage: CALL BISECT(N,EPS1,D,E,E2,LB,UB,MM,M,W,IND,IERR,RV4,RV5) | On-line doc: CALL GAMSDOC BISECT (or @PRT CMLIB*DOC.BISECT/EISPACK) | Access: LIB NBS*CMLIB

BLKTRI Solves block tridiagonal systems of linear algebraic equations arising from the discretisation of separable elliptic partial differential equations. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): I2b4b | Usage: CALL BLKTRI(IFLG,NP,N,AN,BN,CN,MP,M,AM,BM,CM,IDIMY,Y,IERROR, W) | On-line doc: CALL GAMSDOC BLKTRI (or @PRT CMLIB*DOC.BLKTRI/FSHPK) | Tests: CMLIB*TEST-SOURCE.BLKTRI/FSHPK | Access: LIB NBS*CMLIB

BNDACC Introduce new blocks of data for banded least squares problems. See SUBROUTINE BND SOL. | Portable single precision Fortran subprogram in FC sublibrary of CMLIB library. | Class(es): D0 | Usage: CALL BNDACC(G,MDG,NB,IP,IR,MT,JT) | On-line doc: CALL GAMSDOC BNDACC (or @PRT CMLIB*DOC.BNDACC/FC) | Tests: CMLIB*TEST-SOURCE.\$F/FC | Access: LIB NBS*CMLIB | See also: BND SOL

BNDSOL Solves least squares problem $AX=B$ for banded matrices. | Portable single precision Fortran subprogram in FC sublibrary of CMLIB library. | Class(es): D0 | Usage: CALL BNDSOL(MODE,G,MDG,NB,IP,IR,X,N,RNORM) | On-line doc: CALL GAMSDOC BNDSOL (or @PRT CMLIB*DOC.BNDSOL/FC) | Tests: CMLIB*TEST-SOURCE.\$F/FC | Access: LIB NBS*CMLIB

BOXPLOT Prints boxplots - median, hinges, inner and outer fences - for one or more levels. Options: form of plots, notches (confidence interval for population medians). | Command in MINITAB Proprietary interactive system. Class(es): L3d | Usage: BOXPlots for data in C [levels in C] | subcommands LINES = K; NOTCH the boxplots; LEVELS K,...,K [for C]. | On-line doc: HELP BOXPLOT (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

BQR Computes some of the eigenvalues of a real symmetric band matrix using the QR method with shifts of origin. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a6 | Usage: CALL BQR(NM,N,MB,A,T,R,IERR,NV,RV) | On-line doc: CALL GAMSDOC BQR (or @PRT CMLIB*DOC.BQR/EISPACK) | Access: LIB NBS*CMLIB

BQUAD Adaptively integrates functions which have discontinuities in their derivatives. User can specify these points. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DBQUAD. | Class(es): H2a2a1 | Usage: CALL BQUAD (F,N,X,EPS,ANS,ERREST) | On-line doc: CALL GAMSDOC BQUAD (or @PRT PORT*DOC.BQUAD) | Access: LIB NBS*PORT

BRANDOM Generates K pseudo-random numbers from binomial distribution (number of successes in n Bernoulli trials with probability p of success). | Command in MINITAB Proprietary interactive system. Class(es): L6a2 | Usage: BRANDom K binomial experiments with $n = K$, $p = K$, put into C | On-line doc: HELP BRANDOM (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

BSPDR Constructs divided difference table from "B" representation of B-spline for a derivative calculation. | Portable single precision Fortran subprogram in BSPLINE sublibrary of CMLIB library. Double precision version is DBSPDR. | Class(es): E3 | Usage: CALL BSPDR(T,A,N,K,NDERIV,AD) | On-line doc: CALL GAMSDOC BSPDR (or @PRT CMLIB*DOC.BSPDR/BSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/BSPLINE | Access: LIB NBS*CMLIB

BSPEV Calculates the value of a spline and its derivatives at X from its "B" representation. | Portable single precision Fortran subprogram in BSPLINE sublibrary of CMLIB library. Double precision version is DBSPEV. | Class(es): E3 K6 | Usage: CALL BSPEV(T,AD,N,K,NDERIV,X,INEV,SVALUE,WORK) | On-line doc: CALL GAMSDOC BSPEV (or @PRT CMLIB*DOC.BSPEV/BSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/BSPLINE | Access: LIB NBS*CMLIB

BSPL1 Evaluates, at a given set of points in a specified mesh interval, basis splines together with selected orders of derivatives. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DBSPL1. | Class(es): E3 K6 | Usage: CALL BSPL1 (K,T,N,X,NX,ILEFT,ID,NID,BX) | On-line doc: CALL GAMSDOC BSPL1 (or @PRT PORT*DOC.BSPL1) | Access: LIB NBS*PORT

- BSPLD** Evaluates at a given set of points in a specified mesh interval, basis splines and their derivatives. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DBSPLD. | Class(es): E3 K6 | Usage: CALL BSPLD (K,T,N,X,NX,ILEFT,MD,BX) | On-line doc: CALL GAMSDOC BSPLD (or @PRT PORT*DOC.BSPLD) | Access: LIB NBS*PORT
- BSPLI** Obtains the integrals of basis splines, from the left-most mesh point to a specified set of points. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DBSPLI. | Class(es): H2a2a1 E3 K6 | Usage: CALL BSPLI (K,T,N,X,NX,ILEFT,BIX) | On-line doc: CALL GAMSDOC BSPLI (or @PRT PORT*DOC.BSPLI) | Access: LIB NBS*PORT
- BSPLN** Evaluates at a given set of points in a specified mesh interval, all the basis splines which are nonzero in that interval. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DBSPLN. | Class(es): E3 K6 | Usage: CALL BSPLN (K,T,N,X,NX,ILEFT,BX) | On-line doc: CALL GAMSDOC BSPLN (or @PRT PORT*DOC.BSPLN) | Access: LIB NBS*PORT
- BSPPP** Converts from "B" representation of B-spline to piecewise polynomial representation. | Portable single precision Fortran subprogram in BSPLINE sublibrary of CMLIB library. Double precision version is DBSPPP. | Class(es): E3 K6 | Usage: CALL BSPPP(T,A,N,K,LDC,C,XI,LXI,WORK) | On-line doc: CALL GAMSDOC BSPPP (or @PRT CMLIB*DOC.BSPPP/BSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/BSPLINE | Access: LIB NBS*CMLIB
- BSPVD** Calculates value and derivatives of order less than NDERIV of all B-spline basis functions which do not vanish at X. | Portable single precision Fortran subprogram in BSPLINE sublibrary of CMLIB library. Double precision version is DBSPVD. | Class(es): E3 K6 | Usage: CALL BSPVD(T,K,NDERIV,X,ILEFT,LDVNIK,VNIKX,WORK) | On-line doc: CALL GAMSDOC BSPVD (or @PRT CMLIB*DOC.BSPVD/BSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/BSPLINE | Access: LIB NBS*CMLIB
- BSPVN** Calculates the value of all (possibly) nonzero B-spline basis functions at X of a given order. | Portable single precision Fortran subprogram in BSPLINE sublibrary of CMLIB library. Double precision version is DBSPVN. | Class(es): E3 K6 | Usage: CALL BSPVN(T,JHIGH,K,INDEX,X,ILEFT,VNIKX,WORK,IWORK) | On-line doc: CALL GAMSDOC BSPVN (or @PRT CMLIB*DOC.BSPVN/BSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/BSPLINE | Access: LIB NBS*CMLIB
- BSQAD** Computes the integral of a B-spline from X1 to X2. The B-spline must be in "B" representation. | Portable single precision Fortran subprogram in BSPLINE sublibrary of CMLIB library. Double precision version is DBSQAD. | Class(es): H2a2a1 E3 K6 | Usage: CALL BSQAD(T,BCOEF,N,K,X1,X2,BQUAD,WORK) | On-line doc: CALL GAMSDOC BSQAD (or @PRT CMLIB*DOC.BSQAD/BSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/BSPLINE | Access: LIB NBS*CMLIB
- BTRIALS** Generates pseudo-random sequence of K 0's and 1's, with the probability p of a 1. | Command in MINITAB Proprietary interactive system. Class(es): L6a2 | Usage: BTRIALS K Bernoulli trials with p = K, put into C | On-line doc: HELP BTRIALS (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- BURAM** Finds the best uniform rational approximation to a given function on a specified mesh. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DBURAM. | Class(es): K2 | Usage: CALL BURAM (NPTS, MESH, FN, M, N, P, Q, DELK) | On-line doc: CALL GAMSDOC BURAM (or @PRT PORT*DOC.BURAM) | Access: LIB NBS*PORT | See also: TCHBP
- BURG** Computes the coefficients of a finite length causal forward or backward prediction filter and uses both the forward and backward predictions in a symmetric manner to generate the maximum entropy spectrum by means of a Toeplitz recursion. | Portable single precision Fortran subprogram in MAXENTROPY sublibrary of CMLIB library. | Class(es): L10f | Usage: CALL BURG(LX,X,F,B,LA,A,M,S,Y,TABLE) | On-line doc: CALL GAMSDOC BURG (or @PRT CMLIB*DOC.BURG/MAXENTROPY) | Access: LIB NBS*CMLIB
- BURM1** Finds the best uniform rational approximation to a given function on a specified mesh, starting from a given initial approximation. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DBURM1. | Class(es): K2 | Usage: CALL BURM1 (NPTS, MESH, FN, MAXITR, ITOL, M, N, P, Q, DELK) | On-line doc: CALL GAMSDOC BURM1 (or @PRT PORT*DOC.BURM1) | Access: LIB NBS*PORT | See also: TCHBP
- BVALU** Calculates (at X) the value of the IDERIV-th derivative of the B-spline from its "B" representation. | Portable single precision Fortran subprogram in BSPLINE sublibrary of CMLIB library. Double precision version is DBVALU. | Class(es): E3 K6 | Usage: CALL BVALU(T,A,N,K,IDERIV,X,INBV,WORK) | On-line doc: CALL GAMSDOC BVALU (or @PRT CMLIB*DOC.BVALU/BSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/BSPLINE | Access: LIB NBS*CMLIB
- BVSUP** Solves a system of linear two-point boundary value problems using superposition, orthogonalization, and variable step integration. | Portable single precision Fortran subprogram in BVSUP sublibrary of CMLIB library. | Class(es): I1b1 | Usage: CALL BVSUP(Y,NROWY,NCOMP,XPTS,NXPTS,A,NROWA,ALPHA,NIC,B,NROWB,BETA,NFC,IGOFX,REAE,IFLAG,WORK,NDW,IWORK,NDIW,NEQIVP) | On-line doc: CALL GAMSDOC BVSUP (or @PRT CMLIB*DOC.BVSUP/BVSUP) | Access: LIB NBS*CMLIB

C

- C02ADE** All zeros of polynomial, Grant and Hitchin's method, complex coefficients. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C02ADF. | Class(es): F1a1b | Usage: CALL C02ADE (AR, AC, N, REZ, IMZ, TOL, IFAIL) | On-line doc: CALL GAMSDOC C02ADE (or @PRT NAG*DOC.C02ADE) | Access: LIB NBS*NAG
- C02ADF** All zeros of polynomial, Grant and Hitchin's method, complex coefficients. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C02ADE. | Class(es): F1a1b | Usage: CALL C02ADF (AR, AC, N, REZ, IMZ, TOL, IFAIL) | On-line doc: CALL GAMSDOC C02ADF (or @PRT NAG*DOC.C02ADF) | Access: LIB NBS*NAG
- C02AEE** All zeros of polynomial, Grant and Hitchin's method, real coefficients. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C02AEF. | Class(es): F1a1a | Usage: CALL C02AEE (A, N, REZ, IMZ, TOL, IFAIL) | On-line doc: CALL GAMSDOC C02AEE (or @PRT NAG*DOC.C02AEE) | Access: LIB NBS*NAG
- C02AEF** All zeros of polynomial, Grant and Hitchin's method, real coefficients. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C02AEE. | Class(es): F1a1a | Usage: CALL C02AEF (A, N, REZ, IMZ, TOL, IFAIL) | On-line doc: CALL GAMSDOC C02AEF (or @PRT NAG*DOC.C02AEF) | Access: LIB NBS*NAG
- C05ADE** Zero of continuous function of one variable, in a given interval, Bus and Dekker algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C05ADF. | Class(es): F1b | Usage: CALL C05ADE (A, B, EPS, ETA, F, X, IFAIL) | On-line doc: CALL GAMSDOC C05ADE (or @PRT NAG*DOC.C05ADE) | Access: LIB NBS*NAG
- C05ADF** Zero of continuous function of one variable, in a given interval, Bus and Dekker algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C05ADE. | Class(es): F1b | Usage: CALL C05ADF (A, B, EPS, ETA, F, X, IFAIL) | On-line doc: CALL GAMSDOC C05ADF (or @PRT NAG*DOC.C05ADF) | Access: LIB NBS*NAG
- C05AGE** Zero of continuous function of one variable, from a given starting value, search for interval, Bus and Dekker algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C05AGF. | Class(es): F1b | Usage: CALL C05AGE (X, H, EPS, ETA, F, A, B, IFAIL) | On-line doc: CALL GAMSDOC C05AGE (or @PRT NAG*DOC.C05AGE) | Access: LIB NBS*NAG
- C05AGF** Zero of continuous function of one variable, from a given starting value, search for interval, Bus and Dekker algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C05AGE. | Class(es): F1b | Usage: CALL C05AGF (X, H, EPS, ETA, F, A, B, IFAIL) | On-line doc: CALL GAMSDOC C05AGF (or @PRT NAG*DOC.C05AGF) | Access: LIB NBS*NAG
- C05AJE** Zero of continuous function of one variable, from a given starting value, continuation method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C05AJF. | Class(es): F1a2 | Usage: CALL C05AJE (X, EPS, ETA, F, NFMAX, IFAIL) | On-line doc: CALL GAMSDOC C05AJE (or @PRT NAG*DOC.C05AJE) | Access: LIB NBS*NAG
- C05AJF** Zero of continuous function of one variable, from a given starting value, continuation method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C05AJE. | Class(es): F1a2 | Usage: CALL C05AJF (X, EPS, ETA, F, NFMAX, IFAIL) | On-line doc: CALL GAMSDOC C05AJF (or @PRT NAG*DOC.C05AJF) | Access: LIB NBS*NAG
- C05AVE** Zero of continuous function of one variable, search for interval containing zero(reverse communication). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C05AVF. | Class(es): F1b | Usage: CALL C05AVE (X, FX, H, BOUNDL, BOUNDU, Y, C, IND, IFAIL) | On-line doc: CALL GAMSDOC C05AVE (or @PRT NAG*DOC.C05AVE) | Access: LIB NBS*NAG
- C05AVF** Zero of continuous function of one variable, search for interval containing zero(reverse communication). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C05AVE. | Class(es): F1b | Usage: CALL C05AVF (X, FX, H, BOUNDL, BOUNDU, Y, C, IND, IFAIL) | On-line doc: CALL GAMSDOC C05AVF (or @PRT NAG*DOC.C05AVF) | Access: LIB NBS*NAG
- C05AXE** Zero of continuous function of one variable, from a given starting value, continuation method (reverse communication). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C05AXF. | Class(es): F1a2 | Usage: CALL C05AXE (X, FX, TOL, IR, SCALE, C, IND, IFAIL) | On-line doc: CALL GAMSDOC C05AXE (or @PRT NAG*DOC.C05AXE) | Access: LIB NBS*NAG
- C05AXF** Zero of continuous function of one variable, from a given starting value, continuation method (reverse communication). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C05AXE. | Class(es): F1a2 | Usage: CALL C05AXF (X, FX, TOL, IR, SCALE, C, IND, IFAIL) | On-line doc: CALL GAMSDOC C05AXF (or @PRT NAG*DOC.C05AXF) | Access: LIB NBS*NAG
- C05AZE** Zero of continuous function of one variable, in a given interval, Bus and Dekker algorithm (reverse communication). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C05AZF. | Class(es): F1b | Usage: CALL C05AZE (X, Y, FX, TOLX, IR, C, IND, IFAIL) | On-line doc: CALL GAMSDOC C05AZE (or @PRT NAG*DOC.C05AZE) | Access: LIB NBS*NAG
- C05AZF** Zero of continuous function of one variable, in a given interval, Bus and Dekker algorithm (reverse communication). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C05AZE. | Class(es): F1b | Usage: CALL C05AZF (X, Y, FX, TOLX, IR, C, IND, IFAIL) | On-line doc: CALL GAMSDOC C05AZF (or @PRT NAG*DOC.C05AZF) | Access: LIB NBS*NAG
- C05NBE** Easy-to-use routine to find a zero of a system of N nonlinear functions in N variables by a modification of the Powell hybrid method. Derivatives of the function are not required. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C05NBF. | Class(es): F2a | Usage: CALL C05NBE(FCN,N,X,FVEC,XTOL,WA,LWA,IFAIL) | On-line doc: CALL GAMSDOC C05NBE (or @PRT NAG*DOC.C05NBE) | Access: LIB NBS*NAG
- C05NBF** Easy-to-use routine to find a zero of a system of N nonlinear functions in N variables by a modification of the Powell hybrid method. Derivatives of the function are not required. | Proprietary double precision Fortran subprogram in NAG library. Single precision version

- is C05NBE. | Class(es): F2a | Usage: CALL C05NBF(FCN,N,X,FVEC,XTOL,WA,LWA,IFAIL) | On-line doc: CALL GAMSDOC C05NBF (or @PRT NAG*DOC.C05NBF) | Access: LIB NBS*NAG
- C05NCE** Finds a zero of a system of N nonlinear functions in N variables by a modification of Powell's hybrid method. Derivatives of the functions are not required. (Comprehensive version of C05NBE.). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C05NCF. | Class(es): F2a | Usage: CALL C05NCE (FCN,N,X,FVEC,XTOL,MAXFEV,ML,MU,EPSFCN,DIAG,MODE, FACTOR,NPRINT, NFEV FJAC,LDFJAC,R,LR,QTF,W,IFAIL) | On-line doc: CALL GAMSDOC C05NCE (or @PRT NAG*DOC.C05NCE) | Access: LIB NBS*NAG
- C05NCF** Finds a zero of a system of N nonlinear functions in N variables by a modification of Powell's hybrid method. Derivatives of the functions are not required. (Comprehensive version of C05NBF.). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C05NCE. | Class(es): F2a | Usage: CALL C05NCF (FCN,N,X,FVEC,XTOL,MAXFEV,ML,MU,EPSFCN,DIAG,MODE, FACTOR,NPRINT, NFEV FJAC,LDFJAC,R,LR,QTF,W,IFAIL) | On-line doc: CALL GAMSDOC C05NCF (or @PRT NAG*DOC.C05NCF) | Access: LIB NBS*NAG
- C05PBE** Easy-to-use routine to find a zero of a system of N nonlinear functions in N variables by a modification of Powell's hybrid method. The user must provide the Jacobian. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C05PBF. | Class(es): F2a | Usage: CALL C05PBE (FCN,N,X,FVEC,FJAC,LDFJAC,XTOL,WA,LWA,IFAIL) | On-line doc: CALL GAMSDOC C05PBE (or @PRT NAG*DOC.C05PBE) | Access: LIB NBS*NAG
- C05PBF** Easy-to-use routine to find a zero of a system of N nonlinear functions in N variables by a modification of Powell's hybrid method. The user must provide the Jacobian. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C05PBE. | Class(es): F2a | Usage: CALL C05PBF (FCN,N,X,FVEC,FJAC,LDFJAC,XTOL,WA,LWA,IFAIL) | On-line doc: CALL GAMSDOC C05PBF (or @PRT NAG*DOC.C05PBF) | Access: LIB NBS*NAG
- C05PCE** Finds a zero of a system of N nonlinear functions in N variables by a modification of Powell's hybrid method. The user must provide the Jacobian. (Comprehensive version of C05PBE.). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C05PCF. | Class(es): F2a | Usage: CALL C05PCE (FCN,N,X,FVEC,FJAC,LDFJAC,XTOL,MAXFEV,DIAG,MODE, FACTOR,NPRINT, NFEV, NFEV,R,LR,QTF,W,IFAIL) | On-line doc: CALL GAMSDOC C05PCE (or @PRT NAG*DOC.C05PCE) | Access: LIB NBS*NAG
- C05PCF** Finds a zero of a system of N nonlinear functions in N variables by a modification of Powell's hybrid method. The user must provide the Jacobian. (Comprehensive version of C05PBF.). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C05PCE. | Class(es): F2a | Usage: CALL C05PCF (FCN,N,X,FVEC,FJAC,LDFJAC,XTOL,MAXFEV,DIAG,MODE, FACTOR,NPRINT, NFEV, NFEV,R,LR,QTF,W,IFAIL) | On-line doc: CALL GAMSDOC C05PCF (or @PRT NAG*DOC.C05PCF) | Access: LIB NBS*NAG
- C05ZAE** Checks the user-provided Jacobian prior to use in C05PBE or C05PCE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C05ZAF. | Class(es): F3 | Usage: CALL C05ZAE (M,N,X,FVEC,FJAC,LDFJAC,XP,FVECP,MODE,ERR) | On-line doc: CALL GAMSDOC C05ZAE (or @PRT NAG*DOC.C05ZAE) | Access: LIB NBS*NAG
- C05ZAF** Checks the user-provided Jacobian prior to use in C05PBF or C05PCF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C05ZAE. | Class(es): F3 | Usage: CALL C05ZAF (M,N,X,FVEC,FJAC,LDFJAC,XP,FVECP,MODE,ERR) | On-line doc: CALL GAMSDOC C05ZAF (or @PRT NAG*DOC.C05ZAF) | Access: LIB NBS*NAG
- C06ACE** Circular convolution of two real vectors of period 2^*m . | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06ACF. | Class(es): D1a10 J2 | Usage: CALL C06ACE (A, B, C, D, N1, M, ND2, SCALE, ABTRAN) | On-line doc: CALL GAMSDOC C06ACE (or @PRT NAG*DOC.C06ACE) | Access: LIB NBS*NAG
- C06ACF** Circular convolution of two real vectors of period 2^*m . | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06ACE. | Class(es): D1a10 J2 | Usage: CALL C06ACF (A, B, C, D, N1, M, ND2, SCALE, ABTRAN) | On-line doc: CALL GAMSDOC C06ACF (or @PRT NAG*DOC.C06ACF) | Access: LIB NBS*NAG
- C06ADE** Discrete Fourier transform, FFT algorithm, complex data values within a multi- variable transform. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06ADF. | Class(es): J1b | Usage: CALL C06ADE (A, B, N, NV, KS, IW, NIW, W1, NW1, W2, NW2, IFAIL) | On-line doc: CALL GAMSDOC C06ADE (or @PRT NAG*DOC.C06ADE) | Access: LIB NBS*NAG
- C06ADF** Discrete Fourier transform, FFT algorithm, complex data values within a multi- variable transform. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06ADE. | Class(es): J1b | Usage: CALL C06ADF (A, B, N, NV, KS, IW, NIW, W1, NW1, W2, NW2, IFAIL) | On-line doc: CALL GAMSDOC C06ADF (or @PRT NAG*DOC.C06ADF) | Access: LIB NBS*NAG
- C06BAE** Performs Shanks' transformation on a given sequence of real values by means of the Epsilon Algorithm of Wynn. A (possibly unreliable) estimate of the absolute error is also given. An erratic, but often powerful method for accelerating sequences. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06BAF. | Class(es): A7 | Usage: CALL C06BAE(SEQN,NCALL,RESULT,ABSERR,WORK,IWORK,IFAIL) | On-line doc: CALL GAMSDOC C06BAE (or @PRT NAG*DOC.C06BAE) | Access: LIB NBS*NAG
- C06BAF** Performs Shanks' transformation on a given sequence of real values by means of the Epsilon Algorithm of Wynn. A (possibly unreliable) estimate of the absolute error is also given. An erratic, but often powerful method for accelerating sequences. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06BAE. | Class(es): A7 | Usage: CALL C06BAF(SEQN,NCALL,RESULT,ABSERR,WORK,IWORK,IFAIL) | On-line doc: CALL GAMSDOC C06BAF (or @PRT NAG*DOC.C06BAF) | Access: LIB NBS*NAG
- C06DBF** Sum of a Chebyshev series. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06DBF.

- | Class(es): C3a2 | Usage: D = C06DBE (X, C, N, S) | On-line doc: CALL GAMSDOC C06DBE (or @PRT NAG*DOC.C06DBE) | Access: LIB NBS*NAG
- C06DBF** Sum of a Chebyshev series. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06DBE. | Class(es): C3a2 | Usage: D = C06DBF (X, C, N, S) | On-line doc: CALL GAMSDOC C06DBF (or @PRT NAG*DOC.C06DBF) | Access: LIB NBS*NAG
- C06EAE** Discrete Fourier transform, FFT algorithm, no extra workspace, real data values. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06EAF. | Class(es): J1a1 | Usage: CALL C06EAE (X, N, IFAIL) | On-line doc: CALL GAMSDOC C06EAE (or @PRT NAG*DOC.C06EAE) | Access: LIB NBS*NAG
- C06EAF** Discrete Fourier transform, FFT algorithm, no extra workspace, real data values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06EAE. | Class(es): J1a1 | Usage: CALL C06EAF (X, N, IFAIL) | On-line doc: CALL GAMSDOC C06EAF (or @PRT NAG*DOC.C06EAF) | Access: LIB NBS*NAG
- C06EBE** Discrete Fourier transform, FFT algorithm, no extra workspace, complex data values (Hermitian sequence). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06EBF. | Class(es): J1a2 | Usage: CALL C06EBE (X, N, IFAIL) | On-line doc: CALL GAMSDOC C06EBE (or @PRT NAG*DOC.C06EBE) | Access: LIB NBS*NAG
- C06EBF** Discrete Fourier transform, FFT algorithm, no extra workspace, complex data values (Hermitian sequence). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06EBE. | Class(es): J1a2 | Usage: CALL C06EBF (X, N, IFAIL) | On-line doc: CALL GAMSDOC C06EBF (or @PRT NAG*DOC.C06EBF) | Access: LIB NBS*NAG
- C06ECE** Discrete Fourier transform, FFT algorithm, no extra workspace, complex data values (general sequence). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06ECF. | Class(es): J1a2 | Usage: CALL C06ECE (X, Y, N, IFAIL) | On-line doc: CALL GAMSDOC C06ECE (or @PRT NAG*DOC.C06ECE) | Access: LIB NBS*NAG
- C06ECF** Discrete Fourier transform, FFT algorithm, no extra workspace, complex data values (general sequence). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06ECE. | Class(es): J1a2 | Usage: CALL C06ECF (X, Y, N, IFAIL) | On-line doc: CALL GAMSDOC C06ECF (or @PRT NAG*DOC.C06ECF) | Access: LIB NBS*NAG
- C06FAE** Discrete Fourier transform, FFT algorithm, extra workspace for greater speed, real data values. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06FAF. | Class(es): J1a1 | Usage: CALL C06FAE (X, N, WORK, IFAIL) | On-line doc: CALL GAMSDOC C06FAE (or @PRT NAG*DOC.C06FAE) | Access: LIB NBS*NAG
- C06FAF** Discrete Fourier transform, FFT algorithm, extra workspace for greater speed, real data values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06FAE. | Class(es): J1a1 | Usage: CALL C06FAF (X, N, WORK, IFAIL) | On-line doc: CALL GAMSDOC C06FAF (or @PRT NAG*DOC.C06FAF) | Access: LIB NBS*NAG
- C06FBE** Discrete Fourier transform, FFT algorithm, extra workspace for greater speed, complex data values (Hermitian sequence). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06FBF. | Class(es): J1a2 | Usage: CALL C06FBE (X, N, WORK, IFAIL) | On-line doc: CALL GAMSDOC C06FBE (or @PRT NAG*DOC.C06FBE) | Access: LIB NBS*NAG
- C06FBF** Discrete Fourier transform, FFT algorithm, extra workspace for greater speed, complex data values (Hermitian sequence). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06FBE. | Class(es): J1a2 | Usage: CALL C06FBF (X, N, WORK, IFAIL) | On-line doc: CALL GAMSDOC C06FBF (or @PRT NAG*DOC.C06FBF) | Access: LIB NBS*NAG
- C06FCE** Discrete Fourier transform, FFT algorithm, extra workspace for greater speed, complex data values (general sequence). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06FCF. | Class(es): J1a2 | Usage: CALL C06FCE (X, Y, N, WORK, IFAIL) | On-line doc: CALL GAMSDOC C06FCE (or @PRT NAG*DOC.C06FCE) | Access: LIB NBS*NAG
- C06FCF** Discrete Fourier transform, FFT algorithm, extra workspace for greater speed, complex data values (general sequence). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06FCE. | Class(es): J1a2 | Usage: CALL C06FCF (X, Y, N, WORK, IFAIL) | On-line doc: CALL GAMSDOC C06FCF (or @PRT NAG*DOC.C06FCF) | Access: LIB NBS*NAG
- C06GBE** Complex conjugate of complex data values, Hermitian sequence. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06GBF. | Class(es): A4a | Usage: CALL C06GBE (X, N, IFAIL) | On-line doc: CALL GAMSDOC C06GBE (or @PRT NAG*DOC.C06GBE) | Access: LIB NBS*NAG
- C06GBF** Complex conjugate of complex data values, Hermitian sequence. | Proprietary double precision Fortran subprogram in NAG library. | Class(es): A4b | Usage: CALL C06GBF (X, N, IFAIL) | On-line doc: CALL GAMSDOC C06GBF (or @PRT NAG*DOC.C06GBF) | Access: LIB NBS*NAG
- C06GCE** Complex conjugate of complex data values, general sequence. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06GCF. | Class(es): A4a | Usage: CALL C06GCE (Y, N, IFAIL) | On-line doc: CALL GAMSDOC C06GCE (or @PRT NAG*DOC.C06GCE) | Access: LIB NBS*NAG
- C06GCF** Complex conjugate of complex data values, general sequence. | Proprietary double precision Fortran subprogram in NAG library. | Class(es): A4b | Usage: CALL C06GCF (Y, N, IFAIL) | On-line doc: CALL GAMSDOC C06GCF (or @PRT NAG*DOC.C06GCF) | Access: LIB NBS*NAG
- CACOS** Arc cosine of complex argument, $\cos^{-1}(z)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4a | Usage: C=CACOS(Z) | On-line doc: CALL GAMSDOC CACOS (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- CACOSH** Arc hyperbolic cosine of complex argument, $\cosh^{-1}(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4c | Usage: C=CACOSH(Z) | On-line doc: CALL GAMSDOC CACOSH (or @PRT

CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

CARG Argument = theta, in radians of complex number, $z = |z| * e^{i*theta}$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): A4a | Usage: R=CARG(Z) | On-line doc: CALL GAMSDOC CARG (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

CASIN Arc sin of complex argument, $\sin^{-1}(z)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4a | Usage: C=CASIN(Z) | On-line doc: CALL GAMSDOC CASIN (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

CASINH Arc hyperbolic sin of complex argument, $\sinh^{-1}(z)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4c | Usage: C=CASINH(Z) | On-line doc: CALL GAMSDOC CASINH (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

CATAN Arc tangent of complex argument, $\tan^{-1}(z)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4a | Usage: C=CATAN(Z) | On-line doc: CALL GAMSDOC CATAN (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

CATAN2 Quadrant correct arctangent of complex arguments, $\tan^{-1}(z_1/z_2)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4a | Usage: C=CATAN2(Z1,Z2) | On-line doc: CALL GAMSDOC CATAN2 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

CATANH Arc hyperbolic tangent of complex argument, $\tanh^{-1}(z)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4c | Usage: C=CATANH(Z) | On-line doc: CALL GAMSDOC CATANH (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

CAUCDF Computes the cumulative distribution function value for the Cauchy distribution with median = 0 and 75%point = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1c | Usage: CALL CAUCDF(X,CDF) | On-line doc: CALL GAMSDOC CAUCDF (or @PRT DATAPAC*DOC.CAUCDF) | Access: LIB NBS*DATAPAC

CAUPDF Computes the probability density function value for the Cauchy distribution with median = 0 and 75%point = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1c | Usage: CALL CAUPDF(X,PDF) | On-line doc: CALL GAMSDOC CAUPDF (or @PRT DATAPAC*DOC.CAUPDF) | Access: LIB NBS*DATAPAC

CAUPLT Generates a Cauchy probability plot with median = 0 and 75%point = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c4c | Usage: CALL CAUPLT(X,N) | On-line doc: CALL GAMSDOC CAUPLT (or @PRT DATAPAC*DOC.CAUPLT) | Access: LIB NBS*DATAPAC

CAUPPF Computes the percent point function value for the Cauchy distribution with median = 0 and 75%point = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2c | Usage: CALL CAUPPF(P,PPF) | On-line doc: CALL GAMSDOC CAUPPF (or @PRT DATAPAC*DOC.CAUPPF) | Access: LIB NBS*DATAPAC

CAURAN Generates a random sample of size N from the Cauchy distribution with median = 0 and 75%point = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a3 | Usage: CALL CAURAN(N,ISTART,X) | On-line doc: CALL GAMSDOC CAURAN (or @PRT DATAPAC*DOC.CAURAN) | Access: LIB NBS*DATAPAC

CAUSF Computes the sparsity function value for the Cauchy distribution with median = 0 and 75%point = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2c | Usage: CALL CAUSF(P,SF) | On-line doc: CALL GAMSDOC CAUSF (or @PRT DATAPAC*DOC.CAUSF) | Access: LIB NBS*DATAPAC

CAXPY Compute a constant times a vector plus a vector, all complex. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. | Class(es): D1a7 | Usage: CALL CAXPY(N,CA,CX,INCX,CY,INCY) | On-line doc: CALL GAMSDOC CAXPY (or @PRT CMLIB*DOC.CAXPY/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

CBABK2 Forms eigenvectors of complex general matrix from eigenvectors of matrix output from CBAL. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL CBABK2(NM,N,LOW,IGH,SCALE,M,ZR,ZI) | On-line doc: CALL GAMSDOC CBABK2 (or @PRT CMLIB*DOC.CBABK2/EISPACK) | Access: LIB NBS*CMLIB | See also: CBAL

CBAL Balances a complex general matrix and isolates eigenvalues whenever possible. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1a | Usage: CALL CBAL(NM,N,AR,AL,LOW,IGH,SCALE) | On-line doc: CALL GAMSDOC CBAL (or @PRT CMLIB*DOC.CBAL/EISPACK) | Access: LIB NBS*CMLIB

CBETA Beta function of complex arguments, $= (\Gamma(z_1) * \Gamma(z_2)) / \Gamma(z_1 + z_2)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C7b | Usage: C=CBETA(Z1,Z2) | On-line doc: CALL GAMSDOC CBETA (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

CBKTR Solves certain complex block tridiag. systems of lin. eqns. arising from the discretization of separable elliptic partial differential equations. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): I2b4b | Usage: CALL CBKTR(IFLG,NP,N,AN,BN,CN,MP,M,AM,BM,CM,IDIY,Y,IERROR,W) | On-line doc: CALL GAMSDOC CBKTR (or @PRT CMLIB*DOC.CBKTR/FSHPK) | Tests: CMLIB*TEST-SOURCE.CBKTR/FSHPK | Access: LIB NBS*CMLIB

CBNRHO Estimation of the bivariate normal correlation coefficient using a contingency table. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L9b L4e3a14 | Usage: CALL CBNRHO (AP,IA,IB,IRCV,EPS,RHO,VAR,IER) | On-line doc: CALL GAMSDOC CBNRHO (or @PRT IMSL*DOC.CBNRHO) | Access: LIB NBS*IMSL

CBRT Cube root of real number. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double

- precision version is DCBRT. | Class(es): C2 | Usage: R=CBRT(X) | On-line doc: CALL GAMSDOC CBRT (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- CCBRT** Complex cube root of complex argument. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C2 | Usage: C=CCBRT(Z) | On-line doc: CALL GAMSDOC CCBRT (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- CCF** Computes and graphs cross-correlations between two time series. | Command in MINITAB Proprietary interactive system. Class(es): L10g1 | Usage: CCF [with up to K lags] between series in C and C | On-line doc: HELP CCF (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- CCHDC** Compute Cholesky decomposition of complex positive definite matrix with optional pivoting. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1b | Usage: CALL CCHDC(A,LDA,P,WORK,JPVT,JOB,INFO) | On-line doc: CALL GAMSDOC CCHDC (or @PRT CMLIB*DOC.CCHDC/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CCHDD** Downdates Cholesky factorization of positive definite complex matrix. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D7b | Usage: CALL CCHDD(R,LDR,P,X,Z,LDZ,NZ,Y,RHO,C,S,INFO) | On-line doc: CALL GAMSDOC CCHDD (or @PRT CMLIB*DOC.CCHDD/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CCHDC
- CCHEX** Updates Cholesky factorization of positive definite complex matrix. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D7b | Usage: CALL CCHEX(R,LDR,P,K,L,Z,LDZ,NZ,C,S,JOB) | On-line doc: CALL GAMSDOC CCHEX (or @PRT CMLIB*DOC.CCHEX/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CCHDC
- CCHUD** Updates Cholesky factorization of positive definite matrix. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D7b | Usage: CALL CCHUD(R,LDR,P,X,Z,LDZ,NZ,Y,RHO,C,S) | On-line doc: CALL GAMSDOC CCHUD (or @PRT CMLIB*DOC.CCHUD/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CCHDC
- CCOPY** Copy a vector X to a vector Y, both complex. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. | Class(es): D1a5 | Usage: CALL CCOPY(N,CX,INCX,CY,INCY) | On-line doc: CALL GAMSDOC CCOPY (or @PRT CMLIB*DOC.CCOPY/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- CCORRS** Perform cross-correlation analysis between a pair of series. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L10g1 | Usage: CALL CCORRS (Y1, Y2, N, NC, RHO, IRHO) | On-line doc: CALL GAMSDOC CCORRS (or @PRT STATLIB*DOC.CCORRS) | Tests: STATLIB*TEST.DEMO4 | Access: LIB NBS*STATLIB
- CCOSH** Hyperbolic cosine of complex argument, cosh z. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4c | Usage: C=CCOSH(Z) | On-line doc: CALL GAMSDOC CCOSH (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- CCOT** Cotangent of complex argument, cot z. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4a | Usage: C=CCOT(Z) | On-line doc: CALL GAMSDOC CCOT (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- CCSPEC** Computes phase and squared coherency spectra from the Fourier transform of the user-supplied correlation functions, with user-supplied lag window truncation values. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L10g1 | Usage: CALL CCSPEC (RHO1, RHO2, NC, RHOC, NCC, NW, LAGS, Y1, Y2, N, SCRAT, NS) | On-line doc: CALL GAMSDOC CCSPEC (or @PRT STATLIB*DOC.CCSPEC) | Tests: STATLIB*TEST.DEMO4 | Access: LIB NBS*STATLIB
- CDADD** Adds two complex double precision numbers. Each is represented by a double precision array of two elements. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): A4b | Usage: CALL CDADD (A,B,C) | On-line doc: CALL GAMSDOC CDADD (or @PRT PORT*DOC.CDADD) | Access: LIB NBS*PORT
- CDCDOT** Computes complex precision dot product and adds a scalar. Uses double precision accumulation. | Portable single precision Fortran subprogram in XBLAS sublibrary of CMLIB library. | Class(es): D1a4 | Usage: C = CDCDOT(N,CB,CX,INCX,CY,INCY) | On-line doc: CALL GAMSDOC CDCDOT (or @PRT CMLIB*DOC.CDCDOT/XBLAS) | Access: LIB NBS*CMLIB
- CDDIV** Divides two double precision complex numbers. Each is represented by a double precision array of two elements. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): A4b | Usage: CALL CDDIV (A,B,C) | On-line doc: CALL GAMSDOC CDDIV (or @PRT PORT*DOC.CDDIV) | Access: LIB NBS*PORT
- CDDIV** Divides two double precision complex numbers. Each is represented by a double precision array of two elements. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): A4b | Usage: CALL CDDIV (A,B,C) | On-line doc: CALL GAMSDOC CDDIV (or @PRT PORT*DOC.CDDIV) | Access: LIB NBS*PORT
- CDEXP** Computes exp(z) for complex double precision z. Complex double precision numbers are represented as a double precision array of two elements. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): C4b | Usage: CALL CDEXP (X,EXP) | On-line doc: CALL GAMSDOC CDEXP (or @PRT PORT*DOC.CDEXP) | Access: LIB NBS*PORT
- CDLOG** Computes ln(z) for complex double precision z. Complex double precision numbers are represented as a double precision array of two elements. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): C4b | Usage: CALL CDLOG (X,LOG) | On-line doc: CALL GAMSDOC CDLOG (or @PRT PORT*DOC.CDLOG) | Access: LIB NBS*PORT

- CDMUL** Multiplies two double precision complex numbers. Each is represented by a double precision array of two elements. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): A4b | Usage: CALL CDMUL (A,B,C) | On-line doc: CALL GAMSDOC CDMUL (or @PRT PORT*DOC.CDMUL) | Access: LIB NBS*PORT
- CDOTC** Compute complex dot product using conjugated vector components. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. | Class(es): D1a4 | Usage: C = CDOTC(N,CX,INCX,CY,INCY) | On-line doc: CALL GAMSDOC CDOTC (or @PRT CMLIB*DOC.CDOTC/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- CDOTU** Compute complex dot product using unconjugated vector components. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. | Class(es): D1a4 | Usage: C = CDOTU(N,CX,INCX,CY,INCY) | On-line doc: CALL GAMSDOC CDOTU (or @PRT CMLIB*DOC.CDOTU/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- CDRIV1** Numerical integration of complex initial value problems for ordinary differential equations, Gear stiff formulas, Easy to use. | Portable single precision Fortran subprogram in CDRIV sublibrary of CMLIB library. | Class(es): 11a2 11a1b | Usage: CALL CDRIV1(N,T,Y,TOUT,MSTATE,EPS,WORK,LENW) | On-line doc: CALL GAMSDOC CDRIV1 (or @PRT CMLIB*DOC.CDRIV1/CDRIV) | Tests: CMLIB*TEST-SOURCE.\$F/CDRIV | Access: LIB NBS*CMLIB
- CDRIV2** Numerical integration of complex initial value problems for ordinary differential equations, Gear stiff and Adams formulas, root finding. | Portable single precision Fortran subprogram in CDRIV sublibrary of CMLIB library. | Class(es): 11a2 11a1b | Usage: CALL CDRIV2(N,T,Y,F,TOUT,MSTATE,IROOT,EPS,EWT,MINT,WORK,LENW, IWORK,LENIW,G) | On-line doc: CALL GAMSDOC CDRIV2 (or @PRT CMLIB*DOC.CDRIV2/CDRIV) | Tests: CMLIB*TEST-SOURCE.\$F/CDRIV | Access: LIB NBS*CMLIB
- CDRIV3** Numerical integration of complex initial value problems for ODEs, Gear and Adams formulas, Implicit eqs., Sparse Jacobians, root finding. | Portable single precision Fortran subprogram in CDRIV sublibrary of CMLIB library. | Class(es): 11a2 11a1b | Usage: CALL CDRIV3(N,T,Y,F,NSTATE,TOUT,NTASK,IROOT,EPS,EWT,IERROR, MINT,MITER, IMPL,ML, MU,MXORD,HMAX,WORK,LENW,IWORK,LENIW,JACOBN,FA, NDE,MXSTEP,G) | On-line doc: CALL GAMSDOC CDRIV3 (or @PRT CMLIB*DOC.CDRIV3/CDRIV) | Tests: CMLIB*TEST-SOURCE.\$F/CDRIV | Access: LIB NBS*CMLIB
- CDRV** Solves sparse systems of linear algebraic equations by Gaussian elimination without pivoting (compressed storage mode). | Portable single precision Fortran subprogram in YSMP sublibrary of CMLIB library. | Class(es): D2a4 | Usage: CALL CDRV(N,R,C,IC,IA,JA,A,B,Z,NSP,RSP,ESP,PATH,FLAG) | On-line doc: CALL GAMSDOC CDRV (or @PRT CMLIB*DOC.CDRV/YSMP) | Tests: CMLIB*TEST-SOURCE.\$Q2/YSMP | Access: LIB NBS*CMLIB
- CDSUB** Subtracts two double precision complex numbers. Each is represented by a double precision array of two elements. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): A4b | Usage: CALL CDSUB (A,B,C) | On-line doc: CALL GAMSDOC CDSUB (or @PRT PORT*DOC.CDSUB) | Access: LIB NBS*PORT
- CEIL** Finds the smallest integer greater than or equal to x. Input and output are real. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DCEIL. | Class(es): C1 | Usage: X = CEIL (X) | On-line doc: CALL GAMSDOC CEIL (or @PRT PORT*DOC.CEIL) | Access: LIB NBS*PORT
- CENTER** Centers data to mean 0, standard deviation 1. Optionally can select location and scale or minimum and maximum. | Command in MINITAB Proprietary interactive system. Class(es): L2a | Usage: CENTER C,...,C put into C,...,C ; subcommands LOCATION [K,..., K]; SCALE [K,...,K]; MINMAX [K,K]. | On-line doc: HELP CENTER (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- CEXPRL** Relative error exponential from first order, $((e^*z)-1)/z$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4b | Usage: C=CEXPRL(Z) | On-line doc: CALL GAMSDOC CEXPRL (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- CFFTB** Backward complex discrete (fast) Fourier transform. Performs Fourier synthesis. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): J1a2 | Usage: CALL CFFTB(N,C,WSAVE) | On-line doc: CALL GAMSDOC CFFTB (or @PRT CMLIB*DOC.CFFTB/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: CFFTF,CFFTI
- CFFTF** Forward complex discrete (fast) Fourier transform. Performs Fourier analysis. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): J1a2 | Usage: CALL CFFTF(N,C,WSAVE) | On-line doc: CALL GAMSDOC CFFTF (or @PRT CMLIB*DOC.CFFTF/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: CFFTB,CFFTI
- CFFTI** Initialize array WSAVE for forward/back FFT. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): -0- | Usage: CALL CFFTI(N,WSAVE) | On-line doc: CALL GAMSDOC CFFTI (or @PRT CMLIB*DOC.CFFTI/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: CFFTF,CFFTB
- CG** Computes the eigenvalues and, optionally, the eigenvectors of a complex general matrix. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a4 | Usage: CALL CG(NM,N,AR,AI,WR,WI,MATZ,ZR,ZI,FV1,FV2,FV3,IERR) | On-line doc: CALL GAMSDOC CG (or @PRT CMLIB*DOC.CG/EISPACK) | Access: LIB NBS*CMLIB
- CGAMMA** Gamma function of complex argument, Gamma(z). | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C7a | Usage: C=CGAMMA(Z) | On-line doc: CALL GAMSDOC CGAMMA (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- CGAMR** Reciprocal gamma function of complex argument, 1/Gamma(z). | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C7a | Usage: C=CGAMR(Z) | On-line doc: CALL GAMSDOC CGAMR (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

- CGBCO** Compute LU factorization of complex band matrix and estimate its condition. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2c2 | Usage: CALL CGBCO(ABD,LDA,N,ML,MU,IPVT,RCOND,Z) | On-line doc: CALL GAMSDOC CGBCO (or @PRT CMLIB*DOC.CGBCO/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CGBDI** Compute determinant of complex band matrix from its LU factors. (No provision for computing inverse directly.). | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D3c2 | Usage: CALL CGBDI(ABD,LDA,N,ML,MU,IPVT,DET) | On-line doc: CALL GAMSDOC CGBDI (or @PRT CMLIB*DOC.CGBDI/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CGBCO CGBFA
- CGBFA** Compute LU factorization of general complex band matrix. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2c2 | Usage: CALL CGBFA(ABD,LDA,N,ML,MU,IPVT,INFO) | On-line doc: CALL GAMSDOC CGBFA (or @PRT CMLIB*DOC.CGBFA/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CGBSL** Uses LU factorization of complex band matrix to solve systems. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2c2 | Usage: CALL CGBSL(ABD,LDA,N,ML,MU,IPVT,B,JOB) | On-line doc: CALL GAMSDOC CGBSL (or @PRT CMLIB*DOC.CGBSL/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CGBCO CGBFA
- CGECO** Compute LU factorization of general complex matrix and estimate its condition. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2c1 | Usage: CALL CGECO(A,LDA,N,IPVT,RCOND,Z) | On-line doc: CALL GAMSDOC CGECO (or @PRT CMLIB*DOC.CGECO/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CGEDI** Compute determinant and/or inverse of general complex matrix from its LU factors. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2c1 D3c1 | Usage: CALL CGEDI(A,LDA,N,IPVT,DET,WORK,JOB) | On-line doc: CALL GAMSDOC CGEDI (or @PRT CMLIB*DOC.CGEDI/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CGECO CGEFA
- CGEEV** Computes the eigenvalues and, optionally, the eigenvectors of a general complex matrix. | Portable single precision Fortran subprogram in LICEPACK sublibrary of CMLIB library. | Class(es): D4a4 | Usage: CALL CGEEV(A,LDA,N,E,V,LDV,WORK,JOB,INFO) | On-line doc: CALL GAMSDOC CGEEV (or @PRT CMLIB*DOC.CGEEV/LICEPACK) | Tests: CMLIB*TEST-SOURCE.CGEEV/LICEPACK | Access: LIB NBS*CMLIB
- CGEFA** Compute LU factorization of general complex matrix. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2c1 | Usage: CALL CGEFA(A,LDA,N,IPVT,INFO) | On-line doc: CALL GAMSDOC CGEFA (or @PRT CMLIB*DOC.CGEFA/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CGEFS** Factors and solves a general complex system of linear equations. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. | Class(es): D2c1 | Usage: CALL CGEFS(A,LDA,N,V,ITASK,IND,WORK,IWORK) | On-line doc: CALL GAMSDOC CGEFS (or @PRT CMLIB*DOC.CGEFS/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F3/LINDRIVES | Access: LIB NBS*CMLIB
- CGEIR** Factors and solves a general complex system of linear equations and provides estimate of accuracy of the solution (needs NXN extra storage). | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. | Class(es): D2c1 | Usage: CALL CGEIR(A,LDA,N,V,ITASK,IND,WORK,IWORK) | On-line doc: CALL GAMSDOC CGEIR (or @PRT CMLIB*DOC.CGEIR/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F3/LINDRIVES | Access: LIB NBS*CMLIB
- CGESL** Use LU factorization of general complex matrix to solve systems. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2c1 | Usage: CALL CGESL(A,LDA,N,IPVT,B,JOB) | On-line doc: CALL GAMSDOC CGESL (or @PRT CMLIB*DOC.CGESL/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CGECO CGEFA
- CGTSL** Solves systems with general complex tridiagonal matrix. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2c2a | Usage: CALL CGTSL(N,C,D,E,B,INFO) | On-line doc: CALL GAMSDOC CGTSL (or @PRT CMLIB*DOC.CGTSL/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CH** Computes the eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a3 | Usage: CALL CH(NM,N,AR,AI,W,MATZ,ZR,ZI,FV1,FV2,FM1,IERR) | On-line doc: CALL GAMSDOC CH (or @PRT CMLIB*DOC.CH/EISPACK) | Access: LIB NBS*CMLIB
- CHFDV** Evaluates a cubic polynomial and its first derivative at an array of points. The polynomial must be given in Hermite form. | Portable single precision Fortran subprogram in PCHIP sublibrary of CMLIB library. | Class(es): E3 | Usage: CALL CHFDV(X1,X2,F1,F2,D1,D2,NE,XE,FE,DE,NEXT,IERR) | On-line doc: CALL GAMSDOC CHFDV (or @PRT CMLIB*DOC.CHFDV/PCHIP) | Tests: CMLIB*TEST-SOURCE.\$Q/PCHIP | Access: LIB NBS*CMLIB
- CHFEV** Evaluates a cubic polynomial given in Hermite form at an array of points. | Portable single precision Fortran subprogram in PCHIP sublibrary of CMLIB library. | Class(es): E3 | Usage: CALL CHFEV(X1,X2,F1,F2,D1,D2,NE,XE,FE,NEXT,IERR) | On-line doc: CALL GAMSDOC CHFEV (or @PRT CMLIB*DOC.CHFEV/PCHIP) | Tests: CMLIB*TEST-SOURCE.\$Q/PCHIP | Access: LIB NBS*CMLIB

- CHICO** Computes factorization of complex Hermitian indefinite matrix and estimates its condition. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a | Usage: CALL CHICO(A,LDA,N,KPVT,RCOND,Z) | On-line doc: CALL GAMSDOC CHICO (or @PRT CMLIB*DOC.CHICO/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CHIDI** Uses factorization of complex Hermitian indefinite matrix to compute its inertia determinant, and/or inverse. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a D3d1a | Usage: CALL CHIDI(A,LDA,N,KPVT,DET,INERT,WORK,JOB) | On-line doc: CALL GAMSDOC CHIDI (or @PRT CMLIB*DOC.CHIDI/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CHICO CHIFA
- CHIEV** Computes the eigenvalues and, optionally, the eigenvectors of a complex hermitian matrix. | Portable single precision Fortran subprogram in LICEPACK sublibrary of CMLIB library. | Class(es): D4a3 | Usage: CALL CHIEV(A,LDA,N,E,V,LDV,WORK,JOB,INFO) | On-line doc: CALL GAMSDOC CHIEV (or @PRT CMLIB*DOC.CHIEV/LICEPACK) | Tests: CMLIB*TEST-SOURCE.CHIEV/LICEPACK | Access: LIB NBS*CMLIB
- CHIFA** Computes factorization of complex Hermitian indefinite matrix. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a | Usage: CALL CHIFA(A,LDA,N,KPVT,INFO) | On-line doc: CALL GAMSDOC CHIFA (or @PRT CMLIB*DOC.CHIFA/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CHISL** Uses factorization of complex Hermitian indefinite matrix to solve systems. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a | Usage: CALL CHISL(A,LDA,N,KPVT,B) | On-line doc: CALL GAMSDOC CHISL (or @PRT CMLIB*DOC.CHISL/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CHICO CHIFA
- CHISQUARE** Performs chi-square test for association (non-independence) on a two-way table and prints standard results. | Command in MINITAB Proprietary interactive system. Class(es): L9b | Usage: CHISquare test on table stored in columns C,...,C | On-line doc: HELP CHISQUARE (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- CHKDER** Checks gradients of M nonlinear functions in N variables evaluated at a point X for consistency with the functions themselves. A companion subprogram to subprograms SNLS1E and SNLS1. This subprogram can also be used to check the coding of the Jacobian matrix calculation. | Portable single precision Fortran subprogram in SNLS1E sublibrary of CMLIB library. | Class(es): -0- | Usage: CALL CHKDER(M,N,X,FVEC,FJAC,LDFJAC,XP,FVECP,MODE,ERR) | On-line doc: CALL GAMSDOC CHKDER (or @PRT CMLIB*DOC.CHKDER/SNLS1E) | Access: LIB NBS*CMLIB
- CHPCO** Computes factorization of complex Hermitian indefinite matrix stored in packed form and estimates its condition. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a | Usage: CALL CHPCO(AP,N,KPVT,RCOND,Z) | On-line doc: CALL GAMSDOC CHPCO (or @PRT CMLIB*DOC.CHPCO/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CHPDI** Uses factorization of complex Hermitian indefinite matrix stored in packed form to compute its inertia, determinant, and inverse. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a D3d1a | Usage: CALL CHPDI(AP,N,KPVT,DET,INERT,WORK,JOB) | On-line doc: CALL GAMSDOC CHPDI (or @PRT CMLIB*DOC.CHPDI/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CHPCO CHPFA
- CHPFA** Computes factorization of complex Hermitian indefinite matrix stored in packed form. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a | Usage: CALL CHPFA(AP,N,KPVT,INFO) | On-line doc: CALL GAMSDOC CHPFA (or @PRT CMLIB*DOC.CHPFA/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CHPSL** Uses factorization of complex Hermitian indefinite matrix stored in packed form to solve systems. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a | Usage: CALL CHPSL(AP,N,KPVT,B) | On-line doc: CALL GAMSDOC CHPSL (or @PRT CMLIB*DOC.CHPSL/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CHPCO CHPFA
- CHSCDF** Computes the cumulative distribution function value for the chi-squared distribution with degrees of freedom parameter = NU. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1c | Usage: CALL CHSCDF(X,NU,CDF) | On-line doc: CALL GAMSDOC CHSCDF (or @PRT DATAPAC*DOC.CHSCDF) | Access: LIB NBS*DAPAC
- CHSPLT** Generates a chi-squared probability plot with integer degrees of freedom parameter value = NU. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c4c | Usage: CALL CHSPLT(X,N,NU) | On-line doc: CALL GAMSDOC CHSPLT (or @PRT DATAPAC*DOC.CHSPLT) | Access: LIB NBS*DAPAC
- CHSPFF** Computes the percent point function value for the chi-squared distribution with integer degrees of freedom parameter = NU. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2c | Usage: CALL CHSPFF(P,NU,PPF) | On-line doc: CALL GAMSDOC CHSPFF (or @PRT DATAPAC*DOC.CHSPFF) | Access: LIB NBS*DAPAC
- CHSRAN** Generates a random sample of size N from the chi-squared distribution with integer degrees of freedom parameter = NU. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a3 | Usage: CALL CHSRAN(N,NU,ISTART,X) | On-line doc: CALL GAMSDOC CHSRAN (or @PRT DATAPAC*DOC.CHSRAN) | Access: LIB NBS*DAPAC

- CHU** Confluent hypergeometric function, $U(a,b,x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DCHU. | Class(es): C11 | Usage: $R=CHU(A,B,X)$ | On-line doc: CALL GAMSDOC CHU (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- CINVIT** Computes eigenvectors of a complex upper Hessenberg matrix associated with specified eigenvalues using inverse iteration. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c2b | Usage: CALL CINVIT(NM,N,AR,AI,WR,WI,SELECT,MM,M,ZR,ZI,IERR,RM1,RM2, RV1,RV2) | On-line doc: CALL GAMSDOC CINVIT (or @PRT CMLIB*DOC.CINVIT/EISPACK) | Access: LIB NBS*CMLIB
- CJYHBS** Computes $J_{\text{sub } 0}$, $J_{\text{sub } 1}$, $Y_{\text{sub } 0}$, $Y_{\text{sub } 1}$, $H_{\text{sub } 0}$, $H_{\text{sub } 1}$ (Bessel and Struve functions) of complex argument. | Portable single precision Fortran subprogram in AMOSLIB sublibrary of CMLIB library. | Class(es): C10a4 C10e | Usage: CALL CJYHBS(Z,KODE,CJ0,CJ1,CY0,CY1,CH0,CH1) | On-line doc: CALL GAMSDOC CJYHBS (or @PRT CMLIB*DOC.CJYHBS/AMOSLIB) | Access: LIB NBS*CMLIB
- CLBETA** Log Beta of complex arguments, $\ln B(z_1,z_2)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C7b | Usage: $C=CLBETA(Z1,Z2)$ | On-line doc: CALL GAMSDOC CLBETA (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- CLIFE** Life table analysis. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L15 | Usage: CALL CLIFE ((ITYPE, N, IRAD, AGE, A, IPOP, IDTH, L, IDL, DR, Q, SQ, POI, SPOI, E SE, T1, IER) | On-line doc: CALL GAMSDOC CLIFE (or @PRT IMSL*DOC.CLIFE) | Access: LIB NBS*IMSL
- CLINQ** Solves a complex system of linear equations. Coefficient matrix must be input as two real matrices. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DCLINQ. | Class(es): D2c1 | Usage: CALL CLINQ (N,AR,AI,BR,BI,NB,XR,XI) | On-line doc: CALL GAMSDOC CLINQ (or @PRT PORT*DOC.CLINQ) | Access: LIB NBS*PORT
- CLNGAM** Log gamma of complex argument, $\ln \Gamma(z)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C7a | Usage: $C=CLNGAM(Z)$ | On-line doc: CALL GAMSDOC CLNGAM (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- CLNREL** Relative error logarithm of complex argument, $\ln(1+z)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4b | Usage: $C=CLNREL(Z)$ | On-line doc: CALL GAMSDOC CLNREL (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- CLOG10** Common logarithm of complex argument, log to the base 10 of z . | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4b | Usage: $C=CLOG10(Z)$ | On-line doc: CALL GAMSDOC CLOG10 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- CLST2** Finds the least squares solution of a complex linear algebraic system of equations $AX=B$. B may be a matrix. Uses real arithmetic. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DCLST2. | Class(es): D9 | Usage: CALL CLST2 (MDIM,NDIM,M,N,AR,AI,BR,BI,NB,XR,XI) | On-line doc: CALL GAMSDOC CLST2 (or @PRT PORT*DOC.CLST2) | Access: LIB NBS*PORT
- CMGNBN** Solves certain complex block tridiag. systems of lin. eqns. arising from Helmholtz or Poisson eqn in 2 dim. Cartesian coordinates. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): 12b4b | Usage: CALL CMGNBN(NPEROD,N,MPEROD,M,A,B,C,IDIMY,Y,IERROR,W) | On-line doc: CALL GAMSDOC CMGNBN (or @PRT CMLIB*DOC.CMGNNB/FSHPK) | Tests: CMLIB*TEST-SOURCE.CMGNNB/FSHPK | Access: LIB NBS*CMLIB
- CNBCO** Factors a complex band matrix by Gaussian elimination and estimates its condition number. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. | Class(es): D2c2 | Usage: CALL CNBCO(ABE,LDA,N,ML,MU,IPVT,RCOND,Z) | On-line doc: CALL GAMSDOC CNBCO (or @PRT CMLIB*DOC.CNBCO/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F3/LINDRIVES | Access: LIB NBS*CMLIB
- CNBDI** Computes the determinant of a complex band matrix from previously computed factors. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. | Class(es): D3c2 | Usage: CALL CNBDI(ABE,LDA,N,ML,MU,IPVT,DET) | On-line doc: CALL GAMSDOC CNBDI (or @PRT CMLIB*DOC.CNBDI/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F3/LINDRIVES | Access: LIB NBS*CMLIB | See also: CNBCO CNBFA
- CNBFA** Factors a non-symmetric complex band matrix by elimination. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. | Class(es): D2c2 | Usage: CALL CNBFA(ABE,LDA,N,ML,MU,IPVT,INFO) | On-line doc: CALL GAMSDOC CNBFA (or @PRT CMLIB*DOC.CNBFA/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F3/LINDRIVES | Access: LIB NBS*CMLIB
- CNBFS** Factors and solves a general complex band matrix system of linear equations. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. | Class(es): D2c2 | Usage: CALL CNBFS(ABE,LDA,N,ML,MU,V,ITASK,IND,WORK,IWORK) | On-line doc: CALL GAMSDOC CNBFS (or @PRT CMLIB*DOC.CNBFS/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F3/LINDRIVES | Access: LIB NBS*CMLIB
- CNBIR** Factors and solves a general nonsymmetric complex band system of equations and estimates accuracy of the solution (requires $N \times (2ML+MU)$ extra storage). | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. | Class(es): D2c2 | Usage: CALL CNBIR(ABE,LDA,N,ML,MU,V,ITASK,IND,WORK,IWORK) | On-line doc: CALL GAMSDOC CNBIR (or @PRT CMLIB*DOC.CNBIR/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F3/LINDRIVES | Access: LIB NBS*CMLIB
- CNBSL** Solves the nonsymmetric complex band system of equations using factors previously computed. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. | Class(es): D2c2 | Usage: CALL CNBSL(ABE,LDA,N,ML,MU,IPVT,B,JOB) | On-

line doc: CALL GAMSDOC CNBSL (or @PRT CMLIB*DOC.CNBSL/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F3/LINDRIVES
 | Access: LIB NBS*CMLIB | See also: CNBCO CNBFA

CNVBDC Converts values from one vector to another. Backward loop, double precision into complex. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVBDC (N,A,B) | On-line doc: CALL GAMSDOC CNVBDC (or @PRT PORT*DOC.CNVBDC) | Access: LIB NBS*PORT

CNVBDI Converts values from one vector to another. Backward loop, double precision into integer. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVBDI (N,A,B) | On-line doc: CALL GAMSDOC CNVBDI (or @PRT PORT*DOC.CNVBDI) | Access: LIB NBS*PORT

CNVBDR Converts values from one vector to another. Backward loop, double precision into real. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVBDR (N,A,B) | On-line doc: CALL GAMSDOC CNVBDR (or @PRT PORT*DOC.CNVBDR) | Access: LIB NBS*PORT

CNVBIC Converts values from one vector to another. Backward loop, integer into complex. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVBIC (N,A,B) | On-line doc: CALL GAMSDOC CNVBIC (or @PRT PORT*DOC.CNVBIC) | Access: LIB NBS*PORT

CNVBID Converts values from one vector to another. Backward loop, integer to double precision. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVBID (N,A,B) | On-line doc: CALL GAMSDOC CNVBID (or @PRT PORT*DOC.CNVBID) | Access: LIB NBS*PORT

CNVBIR Converts values from one vector to another. Backward loop, integer into real. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVBIR (N,A,B) | On-line doc: CALL GAMSDOC CNVBIR (or @PRT PORT*DOC.CNVBIR) | Access: LIB NBS*PORT

CNVBRC Converts values from one vector to another. Backward loop, real into complex. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVBRC (N,A,B) | On-line doc: CALL GAMSDOC CNVBRC (or @PRT PORT*DOC.CNVBRC) | Access: LIB NBS*PORT

CNVBRD Converts values from one vector to another. Backward loop, real into double precision. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVBRD (N,A,B) | On-line doc: CALL GAMSDOC CNVBRD (or @PRT PORT*DOC.CNVBRD) | Access: LIB NBS*PORT

CNVBRI Converts values from one vector to another. Backward loop, real into integer. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVBRI (N,A,B) | On-line doc: CALL GAMSDOC CNVBRI (or @PRT PORT*DOC.CNVBRI) | Access: LIB NBS*PORT

CNVFDC Converts values from one vector to another. Forward loop, double precision into complex. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVFDC (N,A,B) | On-line doc: CALL GAMSDOC CNVFDC (or @PRT PORT*DOC.CNVFDC) | Access: LIB NBS*PORT

CNVFDI Converts values from one vector to another. Forward loop, double precision into integer. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVFID (N,A,B) | On-line doc: CALL GAMSDOC CNVFID (or @PRT PORT*DOC.CNVFID) | Access: LIB NBS*PORT

CNVFDR Converts values from one vector to another. Forward loop, double precision into real. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVFDR (N,A,B) | On-line doc: CALL GAMSDOC CNVFDR (or @PRT PORT*DOC.CNVFDR) | Access: LIB NBS*PORT

CNVFIC Converts values from one vector to another. Forward loop, integer into complex. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVFIC (N,A,B) | On-line doc: CALL GAMSDOC CNVFIC (or @PRT PORT*DOC.CNVFIC) | Access: LIB NBS*PORT

CNVFID Converts values from one vector to another. Forward loop, integer to double precision. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVFID (N,A,B) | On-line doc: CALL GAMSDOC CNVFID (or @PRT PORT*DOC.CNVFID) | Access: LIB NBS*PORT

CNVFIR Converts values from one vector to another. Forward loop, integer to real. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVFIR (N,A,B) | On-line doc: CALL GAMSDOC CNVFIR (or @PRT PORT*DOC.CNVFIR) | Access: LIB NBS*PORT

CNVFRC Converts values from one vector to another. Forward loop, real into complex. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVFRC (N,A,B) | On-line doc: CALL GAMSDOC CNVFRC (or @PRT PORT*DOC.CNVFRC) | Access: LIB NBS*PORT

CNVFRD Converts values from one vector to another. Forward loop, real into double precision. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVFRD (N,A,B) | On-line doc: CALL GAMSDOC CNVFRD (or @PRT PORT*DOC.CNVFRD) | Access: LIB NBS*PORT

CNVFRI Converts values from one vector to another. Forward loop, real into integer. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVFRI (N,A,B) | On-line doc: CALL GAMSDOC CNVFRI (or @PRT PORT*DOC.CNVFRI) | Access: LIB NBS*PORT

PORT*DOC.CNVFRI) | Access: LIB NBS*PORT

CODE Codes the elements of the input vector X - 1.0 for minimum, 2.0 for next larger, etc. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): N6a1b1 | Usage: CALL CODE(X,N,Y) | On-line doc: CALL GAMSDOC CODE (or @PRT DATAPAC*DOC.CODE) | Access: LIB NBS*DAPAC

COMBAK Forms eigenvectors of complex general matrix from eigenvectors of upper Hessenberg matrix output from COMHES. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL COMBAK(NM,LOW,IGH,AR,AI,INT,M,ZR,ZI) | On-line doc: CALL GAMSDOC COMBAK (or @PRT CMLIB*DOC.COMBAK/EISPACK) | Access: LIB NBS*CMLIB | See also: COMHES

COMHES Reduces complex general matrix to complex upper Hessenberg form using stabilized elementary similarity transformations. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1b2 | Usage: CALL COMHES(NM,N,LOW,IGH,AR,AI,INT) | On-line doc: CALL GAMSDOC COMHES (or @PRT CMLIB*DOC.COMHES/EISPACK) | Access: LIB NBS*CMLIB

COMLR Computes eigenvalues of a complex upper Hessenberg matrix using the modified LR method. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c2b | Usage: CALL COMLR(NM,N,LOW,IGH,HR,HI,WR,WI,IERR) | On-line doc: CALL GAMSDOC COMLR (or @PRT CMLIB*DOC.COMLR/EISPACK) | Access: LIB NBS*CMLIB

COMLR2 Computes eigenvalues and eigenvectors of complex upper Hessenberg matrix using modified LR method. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c2b | Usage: CALL COMLR2(NM,N,LOW,IGH,INT,HR,HI,WR,WI,ZR,ZI,IERR) | On-line doc: CALL GAMSDOC COMLR2 (or @PRT CMLIB*DOC.COMLR2/EISPACK) | Access: LIB NBS*CMLIB

COMQR Computes eigenvalues of complex upper Hessenberg matrix using the QR method. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c2b | Usage: CALL COMQR(NM,N,LOW,IGH,HR,HI,WR,WI,IERR) | On-line doc: CALL GAMSDOC COMQR (or @PRT CMLIB*DOC.COMQR/EISPACK) | Access: LIB NBS*CMLIB

COMQR2 Computes eigenvalues and eigenvectors of complex upper Hessenberg matrix. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c2b | Usage: CALL COMQR2(NM,N,LOW,IGH,ORT,ORTI,HR,HI,WR,WI,ZR,ZI,IERR) | On-line doc: CALL GAMSDOC COMQR2 (or @PRT CMLIB*DOC.COMQR2/EISPACK) | Access: LIB NBS*CMLIB

COPY Copies the contents of the vector X into vector Y. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): D1a5 | Usage: CALL COPY(X,N,Y) | On-line doc: CALL GAMSDOC COPY (or @PRT DATAPAC*DOC.COPY) | Access: LIB NBS*DAPAC

COREL Performs correlation analysis of a multivariate random sample. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L4e1a14 L11 | Usage: CALL COREL (YM, N, M, IYM, SCRAT, NS) | On-line doc: CALL GAMSDOC COREL (or @PRT STATLIB*DOC.COREL) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

CORELS Performs correlation analysis of a multivariate random sample with computed results returned to the user. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L4e1a14 L11 | Usage: CALL CORELS (YM, N, M, IYM, SCRAT, NS, NPRT, R, IR, PR, IPR) | On-line doc: CALL GAMSDOC CORELS (or @PRT STATLIB*DOC.CORELS) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

CORR Computes the sample correlation coefficient between the sets of data in the input vectors X and Y. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1e1b | Usage: CALL CORR(X,Y,N,IWRITE,C) | On-line doc: CALL GAMSDOC CORR (or @PRT DATAPAC*DOC.CORR) | Access: LIB NBS*DAPAC

CORRELATION Calculates the Pearson product moment correlation coefficient between two or more pairs of vectors, handles missing values, and optionally saves results. | Command in MINITAB Proprietary interactive system. | Class(es): L1e1b L1e2 | Usage: CORRELATION coefficients between data in columns C₁,...,C_n [store in M] | On-line doc: HELP CORRELATION (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

CORTB Forms eigenvectors of complex general matrix from eigenvectors of upper Hessenberg matrix output from CORTH. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL CORTB(NM,LOW,IGH,AR,AI,ORT,ORTI,M,ZR,ZI) | On-line doc: CALL GAMSDOC CORTB (or @PRT CMLIB*DOC.CORTB/EISPACK) | Access: LIB NBS*CMLIB | See also: CORTH

CORTH Reduces complex general matrix to complex upper Hessenberg using unitary similarity transformations. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1b2 | Usage: CALL CORTH(NM,N,LOW,IGH,AR,AI,ORT,ORTI) | On-line doc: CALL GAMSDOC CORTH (or @PRT CMLIB*DOC.CORTH/EISPACK) | Access: LIB NBS*CMLIB

COS Cosine of real argument, cos x. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4a | Usage: R=COS(X) | On-line doc: CALL GAMSDOC COS (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

COSDG Computes the cosine of an argument given in degrees. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DCOSDG. | Class(es): C4a | Usage: R=COSDG(X) | On-line doc: CALL GAMSDOC COSDG (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

COSH Computes hyperbolic cosine, cosh(x). | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DCOSH. | Class(es): C4c | Usage: X = COSH (X) | On-line doc: CALL GAMSDOC COSH (or @PRT PORT*DOC.COSH) | Access: LIB NBS*PORT

COSQB Fast Fourier transform of quarter wave data. Computes a sequence from cosine series representation. Fourier synthesis. | Portable

single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): J1a3 | Usage: CALL COSQB(N,X,WSAVE) | On-line doc: CALL GAMSDOC COSQB (or @PRT CMLIB*DOC.COSQB/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: COSQF,COSQI

COSQF Computes fast Fourier transform of quarter wave data. Fourier analysis. Computes coefficients in cosine series with odd wave numbers. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): J1a3 | Usage: CALL COSQF(N,X,WSAVE) | On-line doc: CALL GAMSDOC COSQF (or @PRT CMLIB*DOC.COSQF/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: COSQB,COSQI

COSQI Initialize array WSAVE for SUBROUTINES COSQF and COSQB. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): -0- | Usage: CALL COSQI(N,WSAVE) | On-line doc: CALL GAMSDOC COSQI (or @PRT CMLIB*DOC.COSQI/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: COSQF,COSQB

COST Computes discrete (fast) cosine transform of even sequence X(1). | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): J1a3 | Usage: CALL COST(N,X,WSAVE) | On-line doc: CALL GAMSDOC COST (or @PRT CMLIB*DOC.COST/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: COSTI

COSTI Initialize array WSAVE for SUBROUTINE COST. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): -0- | Usage: CALL COSTI(N,WSAVE) | On-line doc: CALL GAMSDOC COSTI (or @PRT CMLIB*DOC.COSTI/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: COST

COT Cotangent of real argument, cot x. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DCOT. | Class(es): C4a | Usage: R=COT(X) | On-line doc: CALL GAMSDOC COT (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

COUNT Computes the number of observations between XMIN and XMAX (inclusively) in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1d | Usage: CALL COUNT(X,N,XMIN,XMAX,IWRITE,XCOUNT) | On-line doc: CALL GAMSDOC COUNT (or @PRT DATAPAC*DOC.COUNT) | Access: LIB NBS*DAPAC

CPBCO Uses Cholesky algorithm to compute factorization of complex positive definite band matrix and estimates its condition. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d2 | Usage: CALL CPBCO(ABD,LDA,N,M,RCOND,Z,INFO) | On-line doc: CALL GAMSDOC CPBCO (or @PRT CMLIB*DOC.CPBCO/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB

CPBDI Uses factorization of complex positive definite band matrix to compute determinant. (No provision for computing inverse.) | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D3d2 | Usage: CALL CPBDI(ABD,LDA,N,M,DET) | On-line doc: CALL GAMSDOC CPBDI (or @PRT CMLIB*DOC.CPBDI/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CPBCO CPBFA

CPBFA Uses Cholesky algorithm to compute factorization of complex positive definite band matrix. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d2 | Usage: CALL CPBFA(ABD,LDA,N,M,INFO) | On-line doc: CALL GAMSDOC CPBFA (or @PRT CMLIB*DOC.CPBFA/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB

CPBSL Uses factorization of complex positive definite band matrix to solve systems. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d2 | Usage: CALL CPBSL(ABD,LDA,N,M,B) | On-line doc: CALL GAMSDOC CPBSL (or @PRT CMLIB*DOC.CPBSL/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CPBCO CPBFA

CPLOT Prints a scatter diagram which condenses as many as 10 lines of plot into one line and trims extreme x- and y-values. Option: form of output. | Command in MINITAB Proprietary interactive system. Class(es): L3d | Usage: CPLOT y in C vs x in C ; subcommands LINES = K; CHARACTERS = K; XBOUNDS are from K to K; YBOUNDS are from K to K. | On-line doc: HELP CPLOT (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

CPOCO Uses Cholesky algorithm to compute factorization of complex positive definite matrix and estimates its condition. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1b | Usage: CALL CPOCO(A,LDA,N,RCOND,Z,INFO) | On-line doc: CALL GAMSDOC CPOCO (or @PRT CMLIB*DOC.CPOCO/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB

CPODI Uses factorization of complex positive definite matrix to compute its determinant and/or inverse. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1b D3d1b | Usage: CALL CPODI(A,LDA,N,DET,JOB) | On-line doc: CALL GAMSDOC CPODI (or @PRT CMLIB*DOC.CPODI/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CPOCO CPOFA

CPOFA Uses Cholesky algorithm to compute factorization of complex positive definite matrix. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1b | Usage: CALL CPOFA(A,LDA,N,INFO) | On-line doc: CALL GAMSDOC CPOFA (or @PRT CMLIB*DOC.CPOFA/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB

CPOFS Factors and solves positive definite symmetric complex system of linear equations. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. | Class(es): D2d1b | Usage: CALL CPOFS(A,LDA,N,V,ITASK,IND,WORK) | On-line doc: CALL GAMSDOC CPOFS (or @PRT CMLIB*DOC.CPOFS/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F3/LINDRIVES | Access: LIB NBS*CMLIB

- CPOIR** Solves positive definite Hermitian complex system of linear equations and estimates the accuracy of the solution (requires NxN extra storage). | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. | Class(es): D2d1b | Usage: CALL CPOIR(A,LDA,N,V,ITASK,IND,WORK) | On-line doc: CALL GAMSDOC CPOIR (or @PRT CMLIB*DOC.CPOIR/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F3/LINDRIVES | Access: LIB NBS*CMLIB
- CPOLY** Finds the zeros of a polynomial with complex coefficients. Uses two real arrays to represent complex numbers, inconvenient. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DCPOLY. | Class(es): F1a1b | Usage: CALL CPOLY (DEGREE,OPR,OPI,ZEROR,ZEROI) | On-line doc: CALL GAMSDOC CPOLY (or @PRT PORT*DOC.CPOLY) | Access: LIB NBS*PORT
- CPOSL** Uses factorization of complex positive definite matrix to solve systems. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1b | Usage: CALL CPOSL(A,LDA,N,B) | On-line doc: CALL GAMSDOC CPOSL (or @PRT CMLIB*DOC.CPOSL/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CPOCO CPOFA
- CPPCO** Uses Cholesky algorithm to factor complex positive definite matrix stored in packed form. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1b | Usage: CALL CPPCO(AP,N,RCOND,Z,INFO) | On-line doc: CALL GAMSDOC CPPCO (or @PRT CMLIB*DOC.CPPCO/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CPPDI** Uses factorization of complex positive definite matrix stored in packed form to compute determinant and/or inverse. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1b D3d1b | Usage: CALL CPPDI(AP,N,DET,JOB) | On-line doc: CALL GAMSDOC CPPDI (or @PRT CMLIB*DOC.CPPDI/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CPPCO CPPFA
- CPPFA** Uses Cholesky algorithm to factor complex positive definite matrix stored in packed form. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1b | Usage: CALL CPPFA(AP,N,INFO) | On-line doc: CALL GAMSDOC CPPFA (or @PRT CMLIB*DOC.CPPFA/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CPPSL** Uses factorization of complex positive definite matrix stored in packed form to solve systems. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1b | Usage: CALL CPPSL(AP,N,B) | On-line doc: CALL GAMSDOC CPPSL (or @PRT CMLIB*DOC.CPPSL/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CPPCO CPPFA
- CPQR79** Computes all the zeros of a general complex polynomial using eigenvalue methods, requiring NxN storage for Nth degree polynomial. | Portable single precision Fortran subprogram in CPQR79 sublibrary of CMLIB library. | Class(es): F1a1b | Usage: CALL CPQR79(NDEG,COEFF,ROOT,IERR,WORK) | On-line doc: CALL GAMSDOC CPQR79 (or @PRT CMLIB*DOC.CPQR79/CPQR79 and CMLIB*DOC.SUMMARY/CPQR79) | Tests: CMLIB*TEST-SOURCE.\$F/CPQR79 | Access: LIB NBS*CMLIB
- CPSI** Psi (digamma) of complex argument, $\Psi(z) = \Gamma'(z)/\Gamma(z)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C7c | Usage: C=CPSI(Z) | On-line doc: CALL GAMSDOC CPSI (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- CPTSL** Solves systems with complex positive definite tridiagonal matrix. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d2a | Usage: CALL CPTSL(N,D,E,B) | On-line doc: CALL GAMSDOC CPTSL (or @PRT CMLIB*DOC.CPTSL/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CPZERO** Computes all the zeros of a polynomial with complex coefficients. Error bounds are also obtained. Uses Newton's Method for systems. | Portable single precision Fortran subprogram in CPZERO sublibrary of CMLIB library. | Class(es): F1a1b | Usage: CALL CPZERO(N,A,R,T,IFLAG,S) | On-line doc: CALL GAMSDOC CPZERO (or @PRT CMLIB*DOC.SUMMARY/CPZERO and CMLIB*DOC.CPZERO/CPZERO) | Tests: CMLIB*TEST-SOURCE.\$Q/CPZERO | Access: LIB NBS*CMLIB
- CQRDC** Computes QR decomposition of general complex matrix. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D5 | Usage: CALL CQRDC(X,LDX,N,P,QRAUX,JPVT,WORK,JOB) | On-line doc: CALL GAMSDOC CQRDC (or @PRT CMLIB*DOC.CQRDC/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CQRSL** Applies the output of CQRDC to compute coordinate transformations, projections, and least squares solutions (general complex matrix). | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D0 D2c1 | Usage: CALL CQRSL(X,LDX,N,K,QRAUX,Y,QY,QTY,B,RSD,XB,JOB,INFO) | On-line doc: CALL GAMSDOC CQRSL (or @PRT CMLIB*DOC.CQRSL/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CQRDC
- CROTG** Construct Givens plane rotation of complex matrix. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. | Class(es): D1b10 | Usage: CALL CROTG(CA,CB,CC,CS) | On-line doc: CALL GAMSDOC CROTG (or @PRT CMLIB*DOC.CROTG/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- CSCAL** Compute complex constant times complex vector. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. | Class(es): D1a6 | Usage: CALL CSCAL(N,CA,CX,INCX) | On-line doc: CALL GAMSDOC CSCAL (or @PRT CMLIB*DOC.CSCAL/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

- CSEVL** Evaluates an n term series of Chebyshev polynomials at a given point. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DCSEVL. | Class(es): C3a2 | Usage: R=CSEVL(X,CS,N) | On-line doc: CALL GAMSDOC CSEVL (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB | See also: INITS
- CSICO** Computes factorization of complex symmetric indefinite matrix and estimates its condition. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a | Usage: CALL CSICO(A,LDA,N,KPVT,RCOND,Z) | On-line doc: CALL GAMSDOC CSICO (or @PRT CMLIB*DOC.CSICO/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CSIDI** Uses factorization of complex symmetric indefinite matrix to compute its determinant and/or inverse. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a D3d1a | Usage: CALL CSIDI(A,LDA,N,KPVT,DET,INERT,WORK,JOB) | On-line doc: CALL GAMSDOC CSIDI (or @PRT CMLIB*DOC.CSIDI/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CSICO CSIFA
- CSIFA** Computes factorization of complex symmetric indefinite matrix. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a | Usage: CALL CSIFA(A,LDA,N,KPVT,INFO) | On-line doc: CALL GAMSDOC CSIFA (or @PRT CMLIB*DOC.CSIFA/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CSINH** Hyperbolic sine of complex argument, $\sinh z$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4c | Usage: C=CSINH(Z) | On-line doc: CALL GAMSDOC CSINH (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- CSISL** Uses factorization of complex symmetric indefinite matrix to solve systems. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a | Usage: CALL CSISL(A,LDA,N,KPVT,B) | On-line doc: CALL GAMSDOC CSISL (or @PRT CMLIB*DOC.CSISL/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CSICO CSIFA
- CSORT** Sorts a character array in either increasing or decreasing order. Optionally another character array can be carried along. | Portable single precision Fortran subprogram in SSORT sublibrary of CMLIB library. | Class(es): N6a2c | Usage: CALL CSORT(X,Y,N,KFLAG,WORK) | On-line doc: CALL GAMSDOC CSORT (or @PRT CMLIB*DOC.CSORT/SSORT) | Tests: CMLIB*TEST-SOURCE.\$Q/SSORT | Access: LIB NBS*CMLIB
- CSPCO** Computes factorization of complex symmetric indefinite matrix stored in packed form and computes its condition. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a | Usage: CALL CSPCO(AP,N,KPVT,RCOND,Z) | On-line doc: CALL GAMSDOC CSPCO (or @PRT CMLIB*DOC.CSPCO/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, 493F2/LINPACKC | Access: LIB NBS*CMLIB
- CSPDI** Uses factorization of complex symmetric indefinite matrix stored in packed form to compute its determinant and/or inverse. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a D3d1a | Usage: CALL CSPDI(AP,N,KPVT,DET,INERT,WORK,JOB) | On-line doc: CALL GAMSDOC CSPDI (or @PRT CMLIB*DOC.CSPDI/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CSPCO CSPFA
- CSPDI** Finds a numerical approximation to the first derivative at requested points in given input data by using spline interpolation. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DCSPDI. | Class(es): H1 | Usage: CALL CSPDI (X,Y,N,XX,YY,YP,NN) | On-line doc: CALL GAMSDOC CSPDI (or @PRT PORT*DOC.CSPDI) | Access: LIB NBS*PORT
- CSPEC** Computes phase and squared coherency spectra from the Fourier transform of the correlation functions, using the Jenkins and Watts window closing techniques. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L10g1 | Usage: CALL CSPEC (Y1, Y2, N) | On-line doc: CALL GAMSDOC CSPEC (or @PRT STATLIB*DOC.CSPEC) | Tests: STATLIB*TEST.DEMO4 | Access: LIB NBS*STATLIB
- CSPECS** Computes phase and squared coherency spectra from the Fourier transform of the correlation functions, with user-supplied lag window truncation values. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L10g1 | Usage: CALL CSPECS (Y1, Y2, N, NW, LAGS, SCRAT, NS) | On-line doc: CALL GAMSDOC CSPECS (or @PRT STATLIB*DOC.CSPECS) | Tests: STATLIB*TEST.DEMO4 | Access: LIB NBS*STATLIB
- CSPFA** Computes factorization of complex symmetric indefinite matrix stored in packed form. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a | Usage: CALL CSPFA(AP,N,KPVT,INFO) | On-line doc: CALL GAMSDOC CSPFA (or @PRT CMLIB*DOC.CSPFA/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CSPFE** Evaluates a cubic spline function which has already been fit to n input data pairs (x,y) by CSPFI. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DCSPFE. | Class(es): E3 K6 | Usage: CALL CSPFE (X,Y,YP,YPP,N,XX,YY,NN) | On-line doc: CALL GAMSDOC CSPFE (or @PRT PORT*DOC.CSPFE) | Access: LIB NBS*PORT
- CSPFI** Fits a cubic spline function to n input data pairs (x,y) with various endpoint conditions. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DCSPFI. | Class(es): E1a | Usage: CALL CSPFI (X,Y,N,B,Y,YP) | On-line doc: CALL GAMSDOC CSPFI (or @PRT PORT*DOC.CSPFI) | Access: LIB NBS*PORT | See also: CSPFE
- CSPIN** Interpolates at requested points in given input data using a spline approximation-not a least squares fit. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DCSPIN. | Class(es): E1a | Usage: CALL CSPIN (X,Y,N,XX,YY,NN) | On-line doc: CALL GAMSDOC CSPIN (or @PRT PORT*DOC.CSPIN) | Access: LIB NBS*PORT

- CSPQU** Finds the integral of a function defined by pairs (x,y) of input points. The x's can be unequally spaced. Uses spline interpolation. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DCSPQU. | Class(es): H2a1b2 | Usage: CALL CSPQU (X,Y,N,XLOW,XHIGH,ANS) | On-line doc: CALL GAMSDOC CSPQU (or @PRT PORT*DOC.CSPQU) | Access: LIB NBS*PORT
- CSPSL** Uses factorization of complex symmetric indefinite matrix stored in packed form to solve systems. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a | Usage: CALL CSPSL(AP,N,KPVT,B) | On-line doc: CALL GAMSDOC CSPSL (or @PRT CMLIB*DOC.CSPSL/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CSPCO CSPFA
- CSROT** Applies Givens plane rotation to complex matrix. | Portable single precision Fortran subprogram in XBLAS sublibrary of CMLIB library. | Class(es): D1b10 | Usage: CALL CSROT(N,CX,INCX,CY,INCY,C,S) | On-line doc: CALL GAMSDOC CSROT (or @PRT CMLIB*DOC.CSROT/XBLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/XBLAS | Access: LIB NBS*CMLIB
- CSSCAL** Compute real constant times complex vector. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. | Class(es): D1a6 | Usage: CALL CSSCAL(N,SA,CX,INCX) | On-line doc: CALL GAMSDOC CSSCAL (or @PRT CMLIB*DOC.CSSCAL/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- CSVDC** Computes Singular Value Decomposition of general complex matrix. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D6 | Usage: CALL CSVDC(X,LDX,N,P,S,E,U,LDU,V,LDV,WORK,JOB,INFO) | On-line doc: CALL GAMSDOC CSVDC (or @PRT CMLIB*DOC.CSVDC/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CSWAP** Interchange vectors X and Y, both complex. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. | Class(es): D1a5 | Usage: CALL CSWAP(N,CX,INCX,CY,INCY) | On-line doc: CALL GAMSDOC CSWAP (or @PRT CMLIB*DOC.CSWAP/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- CTABLE** Prints a coded two-way table, each cell of which is coded with one character for features MAXIMUM, MINIMUM, or EXTREME, and codes for values between the hinges, between binges and inner fences, between inner and outer fences, and beyond the outer fences. | Command in MINITAB Proprietary interactive system. Class(es): L3d | Usage: CTABLE data in C, row levels in C, columns levels in C [, subcommand MAXIMUM or MINIMUM or EXTREME.] | On-line doc: HELP CTABLE (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- CTAN** Tangent of complex argument, $\tan z$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4a | Usage: C=CTAN(Z) | On-line doc: CALL GAMSDOC CTAN (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- CTANH** Hyperbolic tangent of complex argument, $\tanh z$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4c | Usage: C=CTANH(Z) | On-line doc: CALL GAMSDOC CTANH (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- CTLLF** Log-linear fit of contingency table. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L9c | Usage: CALL CTLLF (A,NVAR,NVAL,NMARG,MARG,IMARG,EPS,MAXIT,AFIT,AMAR, INDEX,DEV,WK,IWK, IER) | On-line doc: CALL GAMSDOC CTLLF (or @PRT IMSL*DOC.CTLLF) | Access: LIB NBS*IMSL
- CTPR** Compute exact probabilities for contingency tables. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L9b | Usage: CALL CTPR (A,IA,IR,IC,PRT,PRE,PCHEK,IWK) | On-line doc: CALL GAMSDOC CTPR (or @PRT IMSL*DOC.CTPR) | Access: LIB NBS*IMSL
- CTRBYC** Analysis of a contingency table. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L9b | Usage: CALL CTRBYC (A,IRC,JRC,IR,IC,STAT,IER) | On-line doc: CALL GAMSDOC CTRBYC (or @PRT IMSL*DOC.CTRBYC) | Access: LIB NBS*IMSL
- CTRCO** Estimates condition of complex triangular matrix. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2c3 | Usage: CALL CTRCO(T,LDT,N,RCOND,Z,JOB) | On-line doc: CALL GAMSDOC CTRCO (or @PRT CMLIB*DOC.CTRCO/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB
- CTRDI** Computes determinant and/or inverse of complex triangular matrix. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2c3 D3c3 | Usage: CALL CTRDI(T,LDT,N,DET,JOB,INFO) | On-line doc: CALL GAMSDOC CTRDI (or @PRT CMLIB*DOC.CTRDI/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CTRCO
- CTRSL** Solves systems with complex triangular matrix. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2c3 | Usage: CALL CTRSL(T,LDT,N,B,JOB,INFO) | On-line doc: CALL GAMSDOC CTRSL (or @PRT CMLIB*DOC.CTRSL/LINPACKC) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKC, CMLIB*TEST-SOURCE.\$F2/LINPACKC | Access: LIB NBS*CMLIB | See also: CTRCO

D

- D01AHE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval suitable for well-behaved integrands. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01AHF. | Class(es): H2a1a1 | Usage: D = D01AHE (A, B, EPSR, NPTS, RELERR, F, NLIMIT, IFAIL) | On-line doc: CALL GAMSDOC D01AHE (or @PRT NAG*DOC.D01AHE) | Access: LIB NBS*NAG
- D01AHF** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval suitable for well-behaved integrands. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01AHE. | Class(es): H2a1a1 | Usage: D = D01AHF (A, B, EPSR, NPTS, RELERR, F, NLIMIT, IFAIL) | On-line doc: CALL GAMSDOC D01AHF (or @PRT NAG*DOC.D01AHF) | Access: LIB NBS*NAG
- D01AJE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval allowing for badly-behaved integrands. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01AJF. | Class(es): H2a1a1 | Usage: CALL D01AJE (F, A, B, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D01AJE (or @PRT NAG*DOC.D01AJE) | Access: LIB NBS*NAG
- D01AJF** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval allowing for badly-behaved integrands. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01AJE. | Class(es): H2a1a1 | Usage: CALL D01AJF (F, A, B, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D01AJF (or @PRT NAG*DOC.D01AJF) | Access: LIB NBS*NAG
- D01AKE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, method suitable for oscillating functions. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01AKF. | Class(es): H2a2a1 | Usage: CALL D01AKE (F, A, B, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D01AKE (or @PRT NAG*DOC.D01AKE) | Access: LIB NBS*NAG
- D01AKF** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, method suitable for oscillating functions. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01AKE. | Class(es): H2a2a1 | Usage: CALL D01AKF (F, A, B, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D01AKF (or @PRT NAG*DOC.D01AKF) | Access: LIB NBS*NAG
- D01ALE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, allowing for singularities at user-specified points. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01ALF. | Class(es): H2a2a1 | Usage: CALL D01ALE (F, A, B, NPTS, POINTS, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D01ALE (or @PRT NAG*DOC.D01ALE) | Access: LIB NBS*NAG
- D01ALF** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, allowing for singularities at user-specified points. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01ALE. | Class(es): H2a2a1 | Usage: CALL D01ALF (F, A, B, NPTS, POINTS, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D01ALF (or @PRT NAG*DOC.D01ALF) | Access: LIB NBS*NAG
- D01AME** Quadrature for one-dimensional integrals, adaptive integration of a function over an infinite or semi-infinite interval. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01AMF. | Class(es): H2a3a1 H2a4a1 | Usage: CALL D01AME (F, BOUND, INF, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D01AME (or @PRT NAG*DOC.D01AME) | Access: LIB NBS*NAG
- D01AMF** Quadrature for one-dimensional integrals, adaptive integration of a function over an infinite or semi-infinite interval. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01AME. | Class(es): H2a3a1 H2a4a1 | Usage: CALL D01AMF (F, BOUND, INF, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D01AMF (or @PRT NAG*DOC.D01AMF) | Access: LIB NBS*NAG
- D01ANE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, weight function $\cos(wx)$ or $\sin(wx)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01ANF. | Class(es): H2a2a1 | Usage: CALL D01ANE (G, A, B, OMEGA, KEY, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D01ANE (or @PRT NAG*DOC.D01ANE) | Access: LIB NBS*NAG
- D01ANF** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, weight function $\cos(wx)$ or $\sin(wx)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01ANE. | Class(es): H2a2a1 | Usage: CALL D01ANF (G, A, B, OMEGA, KEY, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D01ANF (or @PRT NAG*DOC.D01ANF) | Access: LIB NBS*NAG
- D01APE** Adaptive integration of a function of one variable over a finite interval with weight function with algebraico-logarithmic endpoint singularities. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01APF. | Class(es): H2a2a1 | Usage: CALL D01APE (G, A, B, ALFA, BETA, KEY, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D01APE (or @PRT NAG*DOC.D01APE) | Access: LIB NBS*NAG
- D01APF** Adaptive integration of a function of one variable over a finite interval with weight function with algebraico-logarithmic endpoint singularities. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01APE. | Class(es): H2a2a1 | Usage: CALL D01APF (G, A, B, ALFA, BETA, KEY, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D01APF (or @PRT NAG*DOC.D01APF) | Access: LIB NBS*NAG
- D01AQE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, weight function $1/(x-c)$ (Hilbert transform). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01AQF. | Class(es): H2a2a1 J4 | Usage: CALL D01AQE (G, A, B, C, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D01AQE (or @PRT NAG*DOC.D01AQE) | Access: LIB NBS*NAG

- D01AQF** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, weight function $1/(x-c)$ (Hilbert transform). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01AQE. | Class(es): H2a2a1 J4 | Usage: CALL D01AQF (G, A, B, C, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D01AQF (or @PRT NAG*DOC.D01AQF) | Access: LIB NBS*NAG
- D01ARE** Computes definite integral over a finite range to a specified relative or absolute accuracy, using Patterson's method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01ARF. | Class(es): H2a1a1 | Usage: CALL D01ARE(A,B,FUN,RELACC,ABSACC,MAXRUL,IPARM,ACC,ANS,N,ALPHA, IFAIL) | On-line doc: CALL GAMSDOC D01ARE (or @PRT NAG*DOC.D01ARE) | Access: LIB NBS*NAG
- D01ARF** Computes definite integral over a finite range to a specified relative or absolute accuracy, using Patterson's method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01ARE. | Class(es): H2a1a1 | Usage: CALL D01ARF(A,B,FUN,RELACC,ABSACC,MAXRUL,IPARM,ACC,ANS,N,ALPHA, IFAIL) | On-line doc: CALL GAMSDOC D01ARF (or @PRT NAG*DOC.D01ARF) | Access: LIB NBS*NAG
- D01BAE** Quadrature for one-dimensional integrals, Gaussian rule-evaluation. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01BAF. | Class(es): H2a1a2 H2a3a2 H2a4a2 | Usage: D = D01BAE (D01XXX, A, B, N, FUN, IFAIL) | On-line doc: CALL GAMSDOC D01BAE (or @PRT NAG*DOC.D01BAE) | Access: LIB NBS*NAG
- D01BAF** Quadrature for one-dimensional integrals, Gaussian rule-evaluation. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01BAE. | Class(es): H2a1a2 H2a3a2 H2a4a2 | Usage: D = D01BAF (D01XXX, A, B, N, FUN, IFAIL) | On-line doc: CALL GAMSDOC D01BAF (or @PRT NAG*DOC.D01BAF) | Access: LIB NBS*NAG
- D01BBE** Weights and abscissae for Gaussian quadrature rules, restricted choice of rule, using pre-computed weights and abscissae. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01BBF. | Class(es): H2c | Usage: CALL D01BBE (D01XXX, A, B, ITYPE, N, WEIGHT, ABCIS, IFAIL) | On-line doc: CALL GAMSDOC D01BBE (or @PRT NAG*DOC.D01BBE) | Access: LIB NBS*NAG
- D01BBF** Weights and abscissae for Gaussian quadrature rules, restricted choice of rule, using pre-computed weights and abscissae. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01BBE. | Class(es): H2c | Usage: CALL D01BBF (D01XXX, A, B, ITYPE, N, WEIGHT, ABCIS, IFAIL) | On-line doc: CALL GAMSDOC D01BBF (or @PRT NAG*DOC.D01BBF) | Access: LIB NBS*NAG
- D01BCE** Weights and abscissae for Gaussian quadrature rules, more general choice of rule calculating the weights and abscissae. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01BCF. | Class(es): H2c | Usage: CALL D01BCE (ITYPE, A, B, C, D, N, WEIGHT, ABCIS, IFAIL) | On-line doc: CALL GAMSDOC D01BCE (or @PRT NAG*DOC.D01BCE) | Access: LIB NBS*NAG
- D01BCF** Weights and abscissae for Gaussian quadrature rules, more general choice of rule calculating the weights and abscissae. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01BCE. | Class(es): H2c | Usage: CALL D01BCF (ITYPE, A, B, C, D, N, WEIGHT, ABCIS, IFAIL) | On-line doc: CALL GAMSDOC D01BCF (or @PRT NAG*DOC.D01BCF) | Access: LIB NBS*NAG
- D01BDE** Quadrature for one-dimensional integrals, non-adaptive integration over a finite interval. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01BDF. | Class(es): H2a1a1 | Usage: CALL D01BDE (F, A, B, EPSABS, EPSREL, RESULT, ABSERR) | On-line doc: CALL GAMSDOC D01BDE (or @PRT NAG*DOC.D01BDE) | Access: LIB NBS*NAG
- D01BDF** Quadrature for one-dimensional integrals, non-adaptive integration over a finite interval. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01BDE. | Class(es): H2a1a1 | Usage: CALL D01BDF (F, A, B, EPSABS, EPSREL, RESULT, ABSERR) | On-line doc: CALL GAMSDOC D01BDF (or @PRT NAG*DOC.D01BDF) | Access: LIB NBS*NAG
- D01DAE** Quadrature for two-dimensional integrals over a finite region. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01DAF. | Class(es): H2b2a1 | Usage: CALL D01DAE (YA, YB, PHI1, PHI2, ABSACC, ANS NPTS, IFAIL) | On-line doc: CALL GAMSDOC D01DAE (or @PRT NAG*DOC.D01DAE) | Access: LIB NBS*NAG
- D01DAF** Quadrature for two-dimensional integrals over a finite region. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01DAE. | Class(es): H2b2a1 | Usage: CALL D01DAF (YA, YB, PHI1, PHI2, ABSACC, ANS NPTS, IFAIL) | On-line doc: CALL GAMSDOC D01DAF (or @PRT NAG*DOC.D01DAF) | Access: LIB NBS*NAG
- D01FAE** Quadrature for multi-dimensional integrals over a hyper-rectangle, Monte Carlo method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01FAF. | Class(es): H2b1a1 | Usage: CALL D01FAE (NDIM, AA, BB, MINPTS, MAXPTS, FUNCTN, EPS, ACC, LENWRK, WRKSTR, FINVAL, IFAIL) | On-line doc: CALL GAMSDOC D01FAE (or @PRT NAG*DOC.D01FAE) | Access: LIB NBS*NAG
- D01FAF** Quadrature for multi-dimensional integrals over a hyper-rectangle, Monte Carlo method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01FAE. | Class(es): H2b1a1 | Usage: CALL D01FAF (NDIM, AA, BB, MINPTS, MAXPTS, FUNCTN, EPS, ACC, LENWRK, WRKSTR, FINVAL, IFAIL) | On-line doc: CALL GAMSDOC D01FAF (or @PRT NAG*DOC.D01FAF) | Access: LIB NBS*NAG
- D01FBE** Quadrature for multi-dimensional integrals over a hyper-rectangle, Gaussian rule-evaluation. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01FBF. | Class(es): H2b1a2 | Usage: D = D01FBE (NDIM, NPTVEC, LWA, WEIGHT, ABCIS, FUN, IFAIL) | On-line doc: CALL GAMSDOC D01FBE (or @PRT NAG*DOC.D01FBE) | Access: LIB NBS*NAG

- D01FBF** Quadrature for multi-dimensional integrals over a hyper-rectangle, Gaussian rule-evaluation. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01FBE. | Class(es): H2b1a2 | Usage: D = D01FBF (NDIM, NPTVEC, LWA, WEIGHT, ABSIS, FUN, IFAIL) | On-line doc: CALL GAMSDOC D01FBF (or @PRT NAG*DOC.D01FBF) | Access: LIB NBS*NAG
- D01FCE** Quadrature for multi-dimensional integrals over a hyper-rectangle, adaptive method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01FCF. | Class(es): H2b1a1 | Usage: CALL D01FCE (NDIM, A, B, MINPTS, MAXPTS, FUNCTN, EPS, ACC, LENWRK, WRKSTR, FINVAL, IFAIL) | On-line doc: CALL GAMSDOC D01FCE (or @PRT NAG*DOC.D01FCE) | Access: LIB NBS*NAG
- D01FCF** Quadrature for multi-dimensional integrals over a hyper-rectangle, adaptive method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01FCE. | Class(es): H2b1a1 | Usage: CALL D01FCF (NDIM, A, B, MINPTS, MAXPTS, FUNCTN, EPS, ACC, LENWRK, WRKSTR, FINVAL, IFAIL) | On-line doc: CALL GAMSDOC D01FCF (or @PRT NAG*DOC.D01FCF) | Access: LIB NBS*NAG
- D01FDE** Calculates an approximation to a definite integral in up to 30 dimensions, using the method of Sag and Szekeres. The region of integration is an n-sphere, or by built-in transformation via the unit n-cube, any product region. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01FDF. | Class(es): H2b1a2 | Usage: CALL D01FDE(NDIM,FUNCTN,SIGMA,REGION,LIMIT,RO,U,RESULT,NCALS, IFAIL) | On-line doc: CALL GAMSDOC D01FDE (or @PRT NAG*DOC.D01FDE) | Access: LIB NBS*NAG
- D01FDF** Calculates an approximation to a definite integral in up to 30 dimensions, using the method of Sag and Szekeres. The region of integration is an n-sphere, or by built-in transformation via the unit n-cube, any product region. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01FDE. | Class(es): H2b1a2 | Usage: CALL D01FDF(NDIM,FUNCTN,SIGMA,REGION,LIMIT,RO,U,RESULT,NCALS, IFAIL) | On-line doc: CALL GAMSDOC D01FDF (or @PRT NAG*DOC.D01FDF) | Access: LIB NBS*NAG
- D01GAE** Quadrature for one-dimensional integrals, integration of a function defined by data values only. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01GAF. | Class(es): H2a1b2 | Usage: CALL D01GAE (X, Y, N, ANS, ER, IFAIL) | On-line doc: CALL GAMSDOC D01GAE (or @PRT NAG*DOC.D01GAE) | Access: LIB NBS*NAG
- D01GAF** Quadrature for one-dimensional integrals, integration of a function defined by data values only. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01GAE. | Class(es): H2a1b2 | Usage: CALL D01GAF (X, Y, N, ANS, ER, IFAIL) | On-line doc: CALL GAMSDOC D01GAF (or @PRT NAG*DOC.D01GAF) | Access: LIB NBS*NAG
- D01GBE** Calculates an approximation to the integral of a function over a hyper-rectangular region, using a Monte-Carlo method. An approximate relative estimate is also returned. Suitable for low accuracy work. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01GBF. | Class(es): H2b1a1 | Usage: CALL D01GBE(NDIM,A,B,MINCLS,MAXCLS,FUNCTN,EPS,ACC,LENWRK, WRKSTR, FINEST,IFAIL) | On-line doc: CALL GAMSDOC D01GBE (or @PRT NAG*DOC.D01GBE) | Access: LIB NBS*NAG
- D01GBF** Calculates an approximation to the integral of a function over a hyper-rectangular region, using a Monte-Carlo method. An approximate relative estimate is also returned. Suitable for low accuracy work. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01GBE. | Class(es): H2b1a1 | Usage: CALL D01GBF(NDIM,A,B,MINCLS,MAXCLS,FUNCTN,EPS,ACC,LENWRK, WRKSTR, FINEST,IFAIL) | On-line doc: CALL GAMSDOC D01GBF (or @PRT NAG*DOC.D01GBF) | Access: LIB NBS*NAG
- D01GCE** Calculates an approximation to a definite integral in up to 20 dimensions, using the Korobov-Conroy number theoretic method. Returns a simple error estimate by repeating the computation with a different (randomized) set of points. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01GCF. | Class(es): H2b1a2 | Usage: CALL D01GCE(NDIM,FUNCTN,REGION,NPTS,VK,NRAND,ITRANS,RES,ERR, IFAIL) | On-line doc: CALL GAMSDOC D01GCE (or @PRT NAG*DOC.D01GCE) | Access: LIB NBS*NAG | See also: D01GYE, D01GZE
- D01GCF** Calculates an approximation to a definite integral in up to 20 dimensions, using the Korobov-Conroy number theoretic method. Returns a simple error estimate by repeating the computation with a different (randomized) set of points. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01GCE. | Class(es): H2b1a2 | Usage: CALL D01GCF(NDIM,FUNCTN,REGION,NPTS,VK,NRAND,ITRANS,RES,ERR, IFAIL) | On-line doc: CALL GAMSDOC D01GCF (or @PRT NAG*DOC.D01GCF) | Access: LIB NBS*NAG | See also: D01GYF, D01GZF
- D01JAE** Attempts to evaluate an integral over an n-dimensional sphere ($n=2,3,4$), to a user specified absolute or relative accuracy, by means of a modified Sag-Szekeres method. Can handle singularities on the surface or at the center of the sphere. Returns an error estimate. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01JAF. | Class(es): H2b2b1 | Usage: CALL D01JAE(F,NDIM,RADIUS,EPSA,EPSR,METHOD,ICOORD,RESULT, ESTERR,NEVALS,IFAIL) | On-line doc: CALL GAMSDOC D01JAE (or @PRT NAG*DOC.D01JAE) | Access: LIB NBS*NAG
- D01JAF** Attempts to evaluate an integral over an n-dimensional sphere ($n=2,3,4$), to a user specified absolute or relative accuracy, by means of a modified Sag-Szekeres method. Can handle singularities on the surface or at the center of the sphere. Returns an error estimate. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01JAE. | Class(es): H2b2b1 | Usage: CALL D01JAF(F,NDIM,RADIUS,EPSA,EPSR,METHOD,ICOORD,RESULT, ESTERR,NEVALS,IFAIL) | On-line doc: CALL GAMSDOC D01JAF (or @PRT NAG*DOC.D01JAF) | Access: LIB NBS*NAG
- D01PAE** Returns a sequence of approximations to the integral of a function over a multi-dimensional simplex, together with an error estimate for the last approximation. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01PAF. | Class(es): H2b2b2 | Usage: CALL D01PAE(NDIM,VERTEX,IV1,IV2,FUNCTN,MINORD,MAXORD,FINVLS, ESTERR,IFAIL) | On-line doc: CALL GAMSDOC D01PAE (or @PRT NAG*DOC.D01PAE) | Access: LIB NBS*NAG

- D01PAF** Returns a sequence of approximations to the integral of a function over a multi-dimensional simplex, together with an error estimate for the last approximation. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01PAE. | Class(es): H2b2b2 | Usage: CALL D01PAF(NDIM, VERTEX, IV1, IV2, FUNCTN, MINORD, MAXORD, FINVLS, ESTERR, IFAIL) | On-line doc: CALL GAMSDOC D01PAF (or @PRT NAG*DOC.D01PAF) | Access: LIB NBS*NAG
- D02AGE** Solves two point boundary value problem for a system of ODEs using initial value techniques. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02AGF. | Class(es): 11b2 11b3 | Usage: CALL D02AGE(H, ERROR, PARERR, PARAM, C, N, N1, M1, AUX, BCAUX, RAAUX, PRSOL, MAT, COPY, WSPACE, WSPAC1, WSPAC2, IFAIL) | On-line doc: CALL GAMSDOC D02AGE (or @PRT NAG*DOC.D02AGE) | Access: LIB NBS*NAG
- D02AGF** Solves two point boundary value problem for a system of ODEs using initial value techniques. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02AGE. | Class(es): 11b2 11b3 | Usage: CALL D02AGF (H, ERROR, PARERR, PARAM, C, N, N1, M1, AUX, BCAUX, RAAUX, PRSOL, MAT, COPY, WSPACE, WSPAC1, WSPAC2, IFAIL) | On-line doc: CALL GAMSDOC D02AGF (or @PRT NAG*DOC.D02AGF) | Access: LIB NBS*NAG
- D02BAE** Initial value problems for system of ordinary differential equations, (simple driver) Runge-Kutta-Merson method, over a range. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02BAF. | Class(es): 11a1a | Usage: CALL D02BAE (X, XEND, N, Y, TOL, FCN, W, IFAIL) | On-line doc: CALL GAMSDOC D02BAE (or @PRT NAG*DOC.D02BAE) | Access: LIB NBS*NAG
- D02BAF** Initial value problems for system of ordinary differential equations, (simple driver) Runge-Kutta-Merson method, over a range. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02BAE. | Class(es): 11a1a | Usage: CALL D02BAF (X, XEND, N, Y, TOL, FCN, W, IFAIL) | On-line doc: CALL GAMSDOC D02BAF (or @PRT NAG*DOC.D02BAF) | Access: LIB NBS*NAG
- D02BBE** Initial value problems for system of ordinary differential equations, (simple driver) Runge-Kutta-Merson method, over a range with intermediate output. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02BBF. | Class(es): 11a1a | Usage: CALL D02BBE (X, XEND, N, Y, TOL, IRELAB, FCN, OUTPUT, W, IFAIL) | On-line doc: CALL GAMSDOC D02BBE (or @PRT NAG*DOC.D02BBE) | Access: LIB NBS*NAG
- D02BBF** Initial value problems for system of ordinary differential equations, (simple driver) Runge-Kutta-Merson method, over a range with intermediate output. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02BBE. | Class(es): 11a1a | Usage: CALL D02BBF (X, XEND, N, Y, TOL, IRELAB, FCN, OUTPUT, W, IFAIL) | On-line doc: CALL GAMSDOC D02BBF (or @PRT NAG*DOC.D02BBF) | Access: LIB NBS*NAG
- D02BDE** Initial value problems for system of O.D.E.s, (simple driver) Runge-Kutta-Merson method, over a range with global error estimate and stiffness check. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02BDF. | Class(es): 11a1a | Usage: CALL D02BDE (X, XEND, N, Y, TOL, IRELAB, FCN, STIFF, YNORM, W, IW, M, OUTPUT, IFAIL) | On-line doc: CALL GAMSDOC D02BDE (or @PRT NAG*DOC.D02BDE) | Access: LIB NBS*NAG
- D02BDF** Initial value problems for system of O.D.E.s, (simple driver) Runge-Kutta-Merson method, over a range with global error estimate and stiffness check. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02BDE. | Class(es): 11a1a | Usage: CALL D02BDF (X, XEND, N, Y, TOL, IRELAB, FCN, STIFF, YNORM, W, IW, M, OUTPUT, IFAIL) | On-line doc: CALL GAMSDOC D02BDF (or @PRT NAG*DOC.D02BDF) | Access: LIB NBS*NAG
- D02BGE** Initial value problems for system of O.D.E.s, (simple driver) Runge-Kutta-Merson method, until a component of the solution attains a given value. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02BGF. | Class(es): 11a1a | Usage: CALL D02BGE (X, XEND, N, Y, TOL, HMAX, M, VAL, FCN, W, IFAIL) | On-line doc: CALL GAMSDOC D02BGE (or @PRT NAG*DOC.D02BGE) | Access: LIB NBS*NAG
- D02BGF** Initial value problems for system of O.D.E.s, (simple driver) Runge-Kutta-Merson method, until a component of the solution attains a given value. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02BGE. | Class(es): 11a1a | Usage: CALL D02BGF (X, XEND, N, Y, TOL, HMAX, M, VAL, FCN, W, IFAIL) | On-line doc: CALL GAMSDOC D02BGF (or @PRT NAG*DOC.D02BGF) | Access: LIB NBS*NAG
- D02BHE** Initial value problems for system of O.D.E.s, (simple driver) Runge-Kutta-Merson method, until a function of the solution is zero. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02BHF. | Class(es): 11a1a | Usage: CALL D02BHE (X, XEND, N, Y, TOL, IRELAB, HMAX, FCN, G, W, IFAIL) | On-line doc: CALL GAMSDOC D02BHE (or @PRT NAG*DOC.D02BHE) | Access: LIB NBS*NAG
- D02BHF** Initial value problems for system of O.D.E.s, (simple driver) Runge-Kutta-Merson method, until a function of the solution is zero. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02BHE. | Class(es): 11a1a | Usage: CALL D02BHF (X, XEND, N, Y, TOL, IRELAB, HMAX, FCN, G, W, IFAIL) | On-line doc: CALL GAMSDOC D02BHF (or @PRT NAG*DOC.D02BHF) | Access: LIB NBS*NAG
- D02CAE** Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Adams method, over a range. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02CAF. | Class(es): 11a1b | Usage: CALL D02CAE (X, XEND, N, Y, TOL, FCN, W, IFAIL) | On-line doc: CALL GAMSDOC D02CAE (or @PRT NAG*DOC.D02CAE) | Access: LIB NBS*NAG
- D02CAF** Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Adams method, over a range. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02CAE. | Class(es): 11a1b | Usage: CALL D02CAF (X, XEND, N, Y, TOL, FCN, W, IFAIL) | On-line doc: CALL GAMSDOC D02CAF (or @PRT NAG*DOC.D02CAF) | Access: LIB NBS*NAG
- D02CBE** Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Adams method, over a range with

- intermediate output. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02CBF. | Class(es): 11a1b | Usage: CALL D02CBE (X, XEND, N, Y, TOL, IRELAB, FCN, OUTPUT, W, IFAIL) | On-line doc: CALL GAMSDOC D02CBE (or @PRT NAG*DOC.D02CBE) | Access: LIB NBS*NAG
- D02CBF** Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Adams method, over a range with intermediate output. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02CBE. | Class(es): 11a1b | Usage: CALL D02CBF (X, XEND, N, Y, TOL, IRELAB, FCN, OUTPUT, W, IFAIL) | On-line doc: CALL GAMSDOC D02CBF (or @PRT NAG*DOC.D02CBF) | Access: LIB NBS*NAG
- D02CGE** Initial value problems for system of O.D.E.s, (simple driver) variable order and step Adams method, until a component of the solution attains a given value. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02CGF. | Class(es): 11a1b | Usage: CALL D02CGE (X, XEND, N, Y, TOL, HMAX, M, VAL, FCN, W, IFAIL) | On-line doc: CALL GAMSDOC D02CGE (or @PRT NAG*DOC.D02CGE) | Access: LIB NBS*NAG
- D02CGF** Initial value problems for system of O.D.E.s, (simple driver) variable order and step Adams method, until a component of the solution attains a given value. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02CGE. | Class(es): 11a1b | Usage: CALL D02CGF (X, XEND, N, Y, TOL, HMAX, M, VAL, FCN, W, IFAIL) | On-line doc: CALL GAMSDOC D02CGF (or @PRT NAG*DOC.D02CGF) | Access: LIB NBS*NAG
- D02CHE** Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Adams method, until a function of the solution is zero. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02CHF. | Class(es): 11a1b | Usage: CALL D02CHE (X, XEND, N, Y, TOL, IRELAB, HMAX, FCN, G, W, IFAIL) | On-line doc: CALL GAMSDOC D02CHE (or @PRT NAG*DOC.D02CHE) | Access: LIB NBS*NAG
- D02CHF** Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Adams method, until a function of the solution is zero. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02CHE. | Class(es): 11a1b | Usage: CALL D02CHF (X, XEND, N, Y, TOL, IRELAB, HMAX, FCN, G, W, IFAIL) | On-line doc: CALL GAMSDOC D02CHF (or @PRT NAG*DOC.D02CHF) | Access: LIB NBS*NAG
- D02EAE** Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Gear method for stiff systems, over a range. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02EAF. | Class(es): 11a2 | Usage: CALL D02EAE (X, XEND, N, Y, TOL, FCN, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02EAE (or @PRT NAG*DOC.D02EAE) | Access: LIB NBS*NAG
- D02EAF** Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Gear method for stiff systems, over a range. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02EAE. | Class(es): 11a2 | Usage: CALL D02EAF (X, XEND, N, Y, TOL, FCN, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02EAF (or @PRT NAG*DOC.D02EAF) | Access: LIB NBS*NAG
- D02EBE** Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for stiff systems, over a range with intermediate output. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02EBF. | Class(es): 11a2 | Usage: CALL D02EBE (X, XEND, N, Y, TOL, IRELAB, FCN, MPED, PEDERV, OUTPUT, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02EBE (or @PRT NAG*DOC.D02EBE) | Access: LIB NBS*NAG
- D02EBF** Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for stiff systems, over a range with intermediate output. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02EBE. | Class(es): 11a2 | Usage: CALL D02EBF (X, XEND, N, Y, TOL, IRELAB, FCN, MPED, PEDERV, OUTPUT, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02EBF (or @PRT NAG*DOC.D02EBF) | Access: LIB NBS*NAG
- D02EGE** Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for stiff systems, until the solution attains a given value. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02EGF. | Class(es): 11a2 | Usage: CALL D02EGE (X, XEND, N, Y, TOL, HMAX, M, VAL, FCN, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02EGE (or @PRT NAG*DOC.D02EGE) | Access: LIB NBS*NAG
- D02EGF** Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for stiff systems, until the solution attains a given value. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02EGE. | Class(es): 11a2 | Usage: CALL D02EGF (X, XEND, N, Y, TOL, HMAX, M, VAL, FCN, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02EGF (or @PRT NAG*DOC.D02EGF) | Access: LIB NBS*NAG
- D02EHE** Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for stiff systems, until a function of the solution is zero. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02EHF. | Class(es): 11a2 | Usage: CALL D02EHE (X, XEND, N, Y, TOL, IRELAB, HMAX, FCN, MPED, PEDERV, G, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02EHE (or @PRT NAG*DOC.D02EHE) | Access: LIB NBS*NAG
- D02EHF** Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for stiff systems, until a function of the solution is zero. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02EHE. | Class(es): 11a2 | Usage: CALL D02EHF (X, XEND, N, Y, TOL, IRELAB, HMAX, FCN, MPED, PEDERV, G, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02EHF (or @PRT NAG*DOC.D02EHF) | Access: LIB NBS*NAG
- D02GAE** Boundary-value problems for system of O.D.E.s, finite difference technique with deferred correction (Pereyra), simple non-linear problem. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02GAF. | Class(es): 11b2 | Usage: CALL D02GAE (U, V, N, A, B, TOL, FCN, MNP, X, Y, NP, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D02GAE (or @PRT NAG*DOC.D02GAE) | Access: LIB NBS*NAG
- D02GAF** Boundary-value problems for system of O.D.E.s, finite difference technique with deferred correction (Pereyra), simple non-linear problem. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02GAE. | Class(es): 11b2 | Usage: CALL D02GAF (U, V, N, A, B, TOL, FCN, MNP, X, Y, NP, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D02GAF

(or @PRT NAG*DOC.D02GAF) | Access: LIB NBS*NAG

D02GBE Boundary-value problems for system of O.D.E.s, finite difference technique with deferred correction (Pereyra), general linear problem. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02GBF. | Class(es): I1b1 | Usage: CALL D02GBE (A, B, N, TOL, FCNF, FCNG, C, D, GAM, MNP, X, Y, NP, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D02GBE (or @PRT NAG*DOC.D02GBE) | Access: LIB NBS*NAG

D02GBF Boundary-value problems for system of O.D.E.s, finite difference technique with deferred correction (Pereyra), general linear problem. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02GBE. | Class(es): I1b1 | Usage: CALL D02GBF (A, B, N, TOL, FCNF, FCNG, C, D, GAM, MNP, X, Y, NP, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D02GBF (or @PRT NAG*DOC.D02GBF) | Access: LIB NBS*NAG

D02HAE Boundary-value problems for system of O.D.E.s, shooting and matching technique, boundary values to be determined. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02HAF. | Class(es): I1b2 | Usage: CALL D02HAE (U, V, N, A, B, TOL, FCN, SOLN, M1, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02HAE (or @PRT NAG*DOC.D02HAE) | Access: LIB NBS*NAG

D02HAF Boundary-value problems for system of O.D.E.s, shooting and matching technique, boundary values to be determined. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02HAE. | Class(es): I1b2 | Usage: CALL D02HAF (U, V, N, A, B, TOL, FCN, SOLN, M1, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02HAF (or @PRT NAG*DOC.D02HAF) | Access: LIB NBS*NAG

D02HBE Boundary-value problems for system of O.D.E.s, shooting and matching technique, general parameters to be determined. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02HBF. | Class(es): I1b2 | Usage: CALL D02HBE (P, N1, PE, E, N, SOLN, M1, FCN, BC, RANGE, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02HBE (or @PRT NAG*DOC.D02HBE) | Access: LIB NBS*NAG

D02HBF Boundary-value problems for system of O.D.E.s, shooting and matching technique, general parameters to be determined. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02HBE. | Class(es): I1b2 | Usage: CALL D02HBF (P, N1, PE, E, N, SOLN, M1, FCN, BC, RANGE, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02HBF (or @PRT NAG*DOC.D02HBF) | Access: LIB NBS*NAG

D02JAE Solves a regular linear two point boundary value problem for a single n-th order ordinary differential equation by a Chebyshev series using collocation and least squares. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02JAF. | Class(es): I1b1 | Usage: CALL D02JAE (N, CF, BC, X0, X1, K1, KP, C, W, LW, IW, IFAIL) | On-line doc: CALL GAMSDOC D02JAE (or @PRT NAG*DOC.D02JAE) | Access: LIB NBS*NAG

D02JAF Solves a regular linear two point boundary value problem for a single n-th order ordinary differential equation by a Chebyshev series using collocation and least squares. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02JAE. | Class(es): I1b1 | Usage: CALL D02JAF (N, CF, BC, X0, X1, K1, KP, C, W, LW, IW, IFAIL) | On-line doc: CALL GAMSDOC D02JAF (or @PRT NAG*DOC.D02JAF) | Access: LIB NBS*NAG

D02JBE Boundary-value problems for system of O.D.E.s, collocation and least-squares, system of 1st order linear equations. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02JBF. | Class(es): I1b1 | Usage: CALL D02JBE (N, CF, BC, X0, X1, K1, KP, C, IC, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D02JBE (or @PRT NAG*DOC.D02JBE) | Access: LIB NBS*NAG

D02JBF Boundary-value problems for system of O.D.E.s, collocation and least-squares, system of 1st order linear equations. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02JBE. | Class(es): I1b1 | Usage: CALL D02JBF (N, CF, BC, X0, X1, K1, KP, C, IC, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D02JBF (or @PRT NAG*DOC.D02JBF) | Access: LIB NBS*NAG

D02KAE Second-order Sturm-Liouville problems, regular system, finite range, eigenvalue only. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02KAF. | Class(es): I1b3 | Usage: CALL D02KAE (XL, XR, COEFFN, BCOND, K, TOL, ELAM, DELAM, MONIT, IFAIL) | On-line doc: CALL GAMSDOC D02KAE (or @PRT NAG*DOC.D02KAE) | Access: LIB NBS*NAG

D02KAF Second-order Sturm-Liouville problems, regular system, finite range, eigenvalue only. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02KAE. | Class(es): I1b3 | Usage: CALL D02KAF (XL, XR, COEFFN, BCOND, K, TOL, ELAM, DELAM, MONIT, IFAIL) | On-line doc: CALL GAMSDOC D02KAF (or @PRT NAG*DOC.D02KAF) | Access: LIB NBS*NAG

D02KDE Second-order Sturm-Liouville problems, regular/singular system, finite/infinite range, eigenvalue only. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02KDF. | Class(es): I1b3 | Usage: CALL D02KDE (XPOINT, M, COEFFN, BDYVAL, K, TOL, ELAM, DELAM, HMAX, MAXIT, MAXFUN MONIT, IFAIL) | On-line doc: CALL GAMSDOC D02KDE (or @PRT NAG*DOC.D02KDE) | Access: LIB NBS*NAG

D02KDF Second-order Sturm-Liouville problems, regular/singular system, finite/infinite range, eigenvalue only. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02KDE. | Class(es): I1b3 | Usage: CALL D02KDF (XPOINT, M, COEFFN, BDYVAL, K, TOL, ELAM, DELAM, HMAX, MAXIT, MAXFUN MONIT, IFAIL) | On-line doc: CALL GAMSDOC D02KDF (or @PRT NAG*DOC.D02KDF) | Access: LIB NBS*NAG

D02KEE Second-order Sturm-Liouville problems, regular/singular system, finite/infinite range, eigenvalue and eigenfunction. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02KEF. | Class(es): I1b3 | Usage: CALL D02KEE (XPOINT, M, MATCH, COEFFN, BDYVAL, K, TOL, ELAM, DELAM, HMAX, MAXIT, MAXFUN, MONIT, REPORT, IFAIL) | On-line doc: CALL GAMSDOC D02KEE (or @PRT NAG*DOC.D02KEE) | Access: LIB NBS*NAG

- D02KEF** Second-order Sturm-Liouville problems, regular/singular system, finite/infinite range, eigenvalue and eigenfunction. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02KEE. | Class(es): 11b3 | Usage: CALL D02KEF (XPOINT, M, MATCH, COEFFN, BDYVAL, K, TOL, ELAM, DELAM, HMAX, MAXIT, MAXFUN, MONIT, REPORT, IFAIL) | On-line doc: CALL GAMSDOC D02KEF (or @PRT NAG*DOC.D02KEF) | Access: LIB NBS*NAG
- D02PAE** Initial value problems for system of O.D.E.s, integrating over a range (facilities for error-control and interrupts) Runge-Kutta-Merson method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02PAF. | Class(es): 11a1a | Usage: CALL D02PAE (X, XEND, N, Y, CIN, TOL, FCN, COMM, CONST, COUT, W, IW, IW1, IFAIL) | On-line doc: CALL GAMSDOC D02PAE (or @PRT NAG*DOC.D02PAE) | Access: LIB NBS*NAG | See also: D02XAE D02XBE
- D02PAF** Initial value problems for system of O.D.E.s, integrating over a range (facilities for error-control and interrupts) Runge-Kutta-Merson method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02PAE. | Class(es): 11a1a | Usage: CALL D02PAF (X, XEND, N, Y, CIN, TOL, FCN, COMM, CONST, COUT, W, IW, IW1, IFAIL) | On-line doc: CALL GAMSDOC D02PAF (or @PRT NAG*DOC.D02PAF) | Access: LIB NBS*NAG | See also: D02XAF D02XBF
- D02QAE** Initial value problems for system of O.D.E.s, integrating over a range (error-control and interrupts) variable-order -step Adams method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02QAF. | Class(es): 11a1b | Usage: CALL D02QAE (X, XEND, N, Y, CIN, TOL, FCN, COMM, CONST, COUT, W, IW, IW1, IFAIL) | On-line doc: CALL GAMSDOC D02QAE (or @PRT NAG*DOC.D02QAE) | Access: LIB NBS*NAG | See also: D02XGE D02XHE
- D02QAF** Initial value problems for system of O.D.E.s, integrating over a range (error-control and interrupts) variable-order -step Adams method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02QAE. | Class(es): 11a1b | Usage: CALL D02QAF (X, XEND, N, Y, CIN, TOL, FCN, COMM, CONST, COUT, W, IW, IW1, IFAIL) | On-line doc: CALL GAMSDOC D02QAF (or @PRT NAG*DOC.D02QAF) | Access: LIB NBS*NAG | See also: D02XGF D02XHF
- D02QBE** Initial value problems for system of O.D.E.s, integrating over a range (error-control and interrupts) variable-order -step Gear method for stiff systems. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02QBF. | Class(es): 11a2 | Usage: CALL D02QBE (X, XEND, N, Y, CIN, TOL, FCN, COMM, CONST, COUT, MPED, PEDERV, PW, W, IW, IW1, IFAIL) | On-line doc: CALL GAMSDOC D02QBE (or @PRT NAG*DOC.D02QBE) | Access: LIB NBS*NAG | See also: D02XGE D02XHE
- D02QBF** Initial value problems for system of O.D.E.s, integrating over a range (error-control and interrupts) variable-order -step Gear method for stiff systems. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02QBE. | Class(es): 11a2 | Usage: CALL D02QBF (X, XEND, N, Y, CIN, TOL, FCN, COMM, CONST, COUT, MPED, PEDERV, PW, W, IW, IW1, IFAIL) | On-line doc: CALL GAMSDOC D02QBF (or @PRT NAG*DOC.D02QBF) | Access: LIB NBS*NAG | See also: D02XGF D02XHF
- D02RAE** Boundary-value problems for system of O.D.E.s, finite difference technique with deferred correction, general non-linear problem, continuation facility. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02RAF. | Class(es): 11b2 | Usage: CALL D02RAE (N, MNP, NP, NUMBEG, NUMMIX, TOL, INIT, X, Y, IY, ABT, FCN, G, IJAC, JACOB, JACOBG, DELEPS, JACEPS, JACGEP, WORK, LWORK, IWORK, LIWORK, IFAIL) | On-line doc: CALL GAMSDOC D02RAE (or @PRT NAG*DOC.D02RAE) | Access: LIB NBS*NAG
- D02RAF** Boundary-value problems for system of O.D.E.s, finite difference technique with deferred correction, general non-linear problem, continuation facility. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02RAE. | Class(es): 11b2 | Usage: CALL D02RAF (N, MNP, NP, NUMBEG, NUMMIX, TOL, INIT, X, Y, IY, ABT, FCN, G, IJAC, JACOB, JACOBG, DELEPS, JACEPS, JACGEP, WORK, LWORK, IWORK, LIWORK, IFAIL) | On-line doc: CALL GAMSDOC D02RAF (or @PRT NAG*DOC.D02RAF) | Access: LIB NBS*NAG
- D02SAE** Boundary-value problems for system of O.D.E.s, shooting and matching technique, general parameters to be determined, subject to extra algebraic equations. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02SAF. | Class(es): 11b2 | Usage: CALL D02SAE (P, M, N, N1, PE, PF, E, DP, NPOINT, WP, IWP, ICOUNT, RANGE, BC, FCN, EQN, CONSTR, YMAX, MONIT, PRSOL, W, IW1, IW2, IFAIL) | On-line doc: CALL GAMSDOC D02SAE (or @PRT NAG*DOC.D02SAE) | Access: LIB NBS*NAG
- D02SAF** Boundary-value problems for system of O.D.E.s, shooting and matching technique, general parameters to be determined, subject to extra algebraic equations. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02SAE. | Class(es): 11b2 | Usage: CALL D02SAF (P, M, N, N1, PE, PF, E, DP, NPOINT, WP, IWP, ICOUNT, RANGE, BC, FCN, EQN, CONSTR, YMAX, MONIT, PRSOL, W, IW1, IW2, IFAIL) | On-line doc: CALL GAMSDOC D02SAF (or @PRT NAG*DOC.D02SAF) | Access: LIB NBS*NAG
- D02TGE** Boundary-value problems for system of ordinary differential equations, collocation and least-squares, system of n-th order linear equations. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02TGF. | Class(es): 11b1 | Usage: CALL D02TGE (N, M, L, X0, X1, K1, KP, C, IC, COEFF, BDYC, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D02TGE (or @PRT NAG*DOC.D02TGE) | Access: LIB NBS*NAG
- D02TGF** Boundary-value problems for system of ordinary differential equations, collocation and least-squares, system of n-th order linear equations. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02TGE. | Class(es): 11b1 | Usage: CALL D02TGF (N, M, L, X0, X1, K1, KP, C, IC, COEFF, BDYC, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D02TGF (or @PRT NAG*DOC.D02TGF) | Access: LIB NBS*NAG
- D02XAE** Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02PAE, all components. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02XAF. | Class(es): 11c | Usage: CALL D02XAE (XSOL, X, COUT, N, Y, W, IW, SOL, IFAIL) | On-line doc: CALL GAMSDOC D02XAE (or @PRT NAG*DOC.D02XAE) | Access: LIB NBS*NAG | See also: D02PAE

- D02XAF** Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02PAF, all components. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02XAE. | Class(es): 11c | Usage: CALL D02XAF (XSOL, X, COUT, N, Y, W, IW, SOL, IFAIL) | On-line doc: CALL GAMSDOC D02XAF (or @PRT NAG*DOC.D02XAF) | Access: LIB NBS*NAG | See also: D02PAF
- D02XBE** Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02PAE, one component. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02XBF. | Class(es): 11c | Usage: CALL D02XBE (XSOL, X, COUT, N, Y, W, IW, M, SOL, IFAIL) | On-line doc: CALL GAMSDOC D02XBE (or @PRT NAG*DOC.D02XBE) | Access: LIB NBS*NAG | See also: D02PAE
- D02XBF** Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02PAF, one component. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02XBE. | Class(es): 11c | Usage: CALL D02XBF (XSOL, X, COUT, N, Y, W, IW, M, SOL, IFAIL) | On-line doc: CALL GAMSDOC D02XBF (or @PRT NAG*DOC.D02XBF) | Access: LIB NBS*NAG | See also: D02PAF
- D02XGE** Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02QAE or D02QBE, all components. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02XGF. | Class(es): 11c | Usage: CALL D02XGE (XSOL, X, CIN, N, W, IW, IW1, SOL, IFAIL) | On-line doc: CALL GAMSDOC D02XGE (or @PRT NAG*DOC.D02XGE) | Access: LIB NBS*NAG | See also: D02QAE D02QBE
- D02XGF** Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02QAF or D02QBF, all components. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02XGE. | Class(es): 11c | Usage: CALL D02XGF (XSOL, X, CIN, N, W, IW, IW1, SOL, IFAIL) | On-line doc: CALL GAMSDOC D02XGF (or @PRT NAG*DOC.D02XGF) | Access: LIB NBS*NAG | See also: D02QAF D02QBF
- D02XHE** Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02QAE or D02QBE, one component. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02XHF. | Class(es): 11c | Usage: CALL D02XHE (XSOL, X, CIN, N, W, IW, IW1, M, SOL, IFAIL) | On-line doc: CALL GAMSDOC D02XHE (or @PRT NAG*DOC.D02XHE) | Access: LIB NBS*NAG | See also: D02QAE D02QBE
- D02XHF** Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02QAF or D02QBF, one component. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02XHE. | Class(es): 11c | Usage: CALL D02XHF (XSOL, X, CIN, N, W, IW, IW1, M, SOL, IFAIL) | On-line doc: CALL GAMSDOC D02XHF (or @PRT NAG*DOC.D02XHF) | Access: LIB NBS*NAG | See also: D02QAF D02QBF
- D02YAE** Initial value problems for system of ordinary differential equations, integration over one step by Runge-Kutta-Merson method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02YAF. | Class(es): 11a1a | Usage: CALL D02YAE (X, H, N, Y, FCN, W, IW1, IW2) | On-line doc: CALL GAMSDOC D02YAE (or @PRT NAG*DOC.D02YAE) | Access: LIB NBS*NAG
- D02YAF** Initial value problems for system of ordinary differential equations, integration over one step by Runge-Kutta-Merson method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02YAE. | Class(es): 11a1a | Usage: CALL D02YAF (X, H, N, Y, FCN, W, IW1, IW2) | On-line doc: CALL GAMSDOC D02YAF (or @PRT NAG*DOC.D02YAF) | Access: LIB NBS*NAG
- D03EAE** Partial differential equations, elliptic, Laplace's equation in 2-d for an arbitrary domain. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D03EAF. | Class(es): 12b1a1b | Usage: CALL D03EAE (STAGE1, EXT, DORM, N, P, Q, X, Y, N1P1, PHI, PHID, ALPHA, C, IC, NP4, ICINT, NP1, IFAIL) | On-line doc: CALL GAMSDOC D03EAE (or @PRT NAG*DOC.D03EAE) | Access: LIB NBS*NAG
- D03EAF** Partial differential equations, elliptic, Laplace's equation in 2-d for an arbitrary domain. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D03EAE. | Class(es): 12b1a1b | Usage: CALL D03EAF (STAGE1, EXT, DORM, N, P, Q, X, Y, N1P1, PHI, PHID, ALPHA, C, IC, NP4, ICINT, NP1, IFAIL) | On-line doc: CALL GAMSDOC D03EAF (or @PRT NAG*DOC.D03EAF) | Access: LIB NBS*NAG
- D03EBE** P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 5-point 2-d molecule, iterate to convergence. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D03EBF. | Class(es): 12b4b | Usage: CALL D03EBE (N1, N2, N1M, A, B, C, D, E, Q, T, APARAM, ITMAX, ITCOUN, ITUSED, NDIR, IXN, IYN, CONRES, CONCHN, RESIDS, CHNGS, WRKSP1, WRKSP2, WRKSP3, IFAIL) | On-line doc: CALL GAMSDOC D03EBE (or @PRT NAG*DOC.D03EBE) | Access: LIB NBS*NAG
- D03EBF** P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 5-point 2-d molecule, iterate to convergence. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D03EBE. | Class(es): 12b4b | Usage: CALL D03EBF (N1, N2, N1M, A, B, C, D, E, Q, T, APARAM, ITMAX, ITCOUN, ITUSED, NDIR, IXN, IYN, CONRES, CONCHN, RESIDS, CHNGS, WRKSP1, WRKSP2, WRKSP3, IFAIL) | On-line doc: CALL GAMSDOC D03EBF (or @PRT NAG*DOC.D03EBF) | Access: LIB NBS*NAG
- D03ECE** P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 7-point 3-d molecule, iterate to convergence. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D03ECF. | Class(es): 12b4b | Usage: CALL D03ECE (N1, N2, N3, N1M, N2M, A, B, C, D, E, F, G, Q, T, APARAM, ITMAX, ITCOUN, ITUSED, NDIR, IXN, IYN, IZN, CONRES, CONCHN, RESIDS, CHNGS, WRKSP1, IFAIL) | On-line doc: CALL GAMSDOC D03ECE (or @PRT NAG*DOC.D03ECE) | Access: LIB NBS*NAG
- D03ECF** P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 7-point 3-d molecule, iterate to convergence. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D03ECE. | Class(es): 12b4b | Usage: CALL D03ECF (N1, N2, N3, N1M, N2M, A, B, C, D, E, F, G, Q, T, APARAM, ITMAX, ITCOUN, ITUSED, NDIR, IXN, IYN, IZN, CONRES, CONCHN, RESIDS, CHNGS, WRKSP1, IFAIL) | On-line doc: CALL GAMSDOC D03ECF (or @PRT NAG*DOC.D03ECF) | Access: LIB NBS*NAG

- D03MAE** Triangulation of a plane region. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D03MAF. | Class(es): 12b4a P2a2c1 | Usage: CALL D03MAE (H, M, N, NB, NPTS, PLACES, INDEX, IDIM, IN, DIST, LD, IFAIL) | On-line doc: CALL GAMSDOC D03MAE (or @PRT NAG*DOC.D03MAE) | Access: LIB NBS*NAG
- D03MAF** Triangulation of a plane region. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D03MAE. | Class(es): 12b4a P2a2c1 | Usage: CALL D03MAF (H, M, N, NB, NPTS, PLACES, INDEX, IDIM, IN, DIST, LD, IFAIL) | On-line doc: CALL GAMSDOC D03MAF (or @PRT NAG*DOC.D03MAF) | Access: LIB NBS*NAG
- D03PAE** P.D.E.s, parabolic, one space variable, method of lines, single equation. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D03PAF. | Class(es): 12a1a | Usage: CALL D03PAE (M, A, B, TS, TOUT, U, NPTS, ACC, WORK, IWK, IND, IFAIL) | On-line doc: CALL GAMSDOC D03PAE (or @PRT NAG*DOC.D03PAE) | Access: LIB NBS*NAG
- D03PAF** P.D.E.s, parabolic, one space variable, method of lines, single equation. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D03PAE. | Class(es): 12a1a | Usage: CALL D03PAF (M, A, B, TS, TOUT, U, NPTS, ACC, WORK, IWK, IND, IFAIL) | On-line doc: CALL GAMSDOC D03PAF (or @PRT NAG*DOC.D03PAF) | Access: LIB NBS*NAG
- D03PBE** P.D.E.s, parabolic, one space variable, method of lines, simple system. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D03PBF. | Class(es): 12a1a | Usage: CALL D03PBE (NPDE, M, PDEF, BNDY, A, B, TS, TOUT, U, NPTS, IMESH, X, ACC, WORK, IWK, IND, IFAIL) | On-line doc: CALL GAMSDOC D03PBE (or @PRT NAG*DOC.D03PBE) | Access: LIB NBS*NAG
- D03PBF** P.D.E.s, parabolic, one space variable, method of lines, simple system. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D03PBE. | Class(es): 12a1a | Usage: CALL D03PBF (NPDE, M, PDEF, BNDY, A, B, TS, TOUT, U, NPTS, IMESH, X, ACC, WORK, IWK, IND, IFAIL) | On-line doc: CALL GAMSDOC D03PBF (or @PRT NAG*DOC.D03PBF) | Access: LIB NBS*NAG
- D03PGE** P.D.E.s, parabolic, one space variable, method of lines, general system. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D03PGF. | Class(es): 12a1a | Usage: CALL D03PGE (NPDE, M, PDEF, BNDY, TS, TOUT, U, IU, NPTS, X, RELERR, ABSERR, INORM, MONTR, IMON, IBAND, WORK, IWK, IND, IFAIL) | On-line doc: CALL GAMSDOC D03PGE (or @PRT NAG*DOC.D03PGE) | Access: LIB NBS*NAG
- D03PGF** P.D.E.s, parabolic, one space variable, method of lines, general system. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D03PGE. | Class(es): 12a1a | Usage: CALL D03PGF (NPDE, M, PDEF, BNDY, TS, TOUT, U, IU, NPTS, X, RELERR, ABSERR, INORM, MONTR, IMON, IBAND, WORK, IWK, IND, IFAIL) | On-line doc: CALL GAMSDOC D03PGF (or @PRT NAG*DOC.D03PGF) | Access: LIB NBS*NAG
- D03UAE** P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 5-point 2-d molecule, one iteration. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D03UAF. | Class(es): 12b4b | Usage: CALL D03UAE (N1, N2, N1M, A, B, C, D, E, APARAM, IT, R, WRKSP1, WRKSP2, IFAIL) | On-line doc: CALL GAMSDOC D03UAE (or @PRT NAG*DOC.D03UAE) | Access: LIB NBS*NAG
- D03UAF** P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 5-point 2-d molecule, one iteration. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D03UAE. | Class(es): 12b4b | Usage: CALL D03UAF (N1, N2, N1M, A, B, C, D, E, APARAM, IT, R, WRKSP1, WRKSP2, IFAIL) | On-line doc: CALL GAMSDOC D03UAF (or @PRT NAG*DOC.D03UAF) | Access: LIB NBS*NAG
- D03UBE** P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 7-point 3-d molecule, one iteration. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D03UBF. | Class(es): 12b4b | Usage: CALL D03UBE (N1, N2, N3, N1M, N2M, A, B, C, D, E, F, G, APARAM, IT, R, WRKSP1, WRKSP2, WRKSP3, IFAIL) | On-line doc: CALL GAMSDOC D03UBE (or @PRT NAG*DOC.D03UBE) | Access: LIB NBS*NAG
- D03UBF** P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 7-point 3-d molecule, one iteration. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D03UBE. | Class(es): 12b4b | Usage: CALL D03UBF (N1, N2, N3, N1M, N2M, A, B, C, D, E, F, G, APARAM, IT, R, WRKSP1, WRKSP2, WRKSP3, IFAIL) | On-line doc: CALL GAMSDOC D03UBF (or @PRT NAG*DOC.D03UBF) | Access: LIB NBS*NAG
- D04AAE** Numerical differentiation of a function of one real variable, derivatives up to order 14. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D04AAF. | Class(es): H1 | Usage: CALL D04AAE (XVAL, NDER, HBASE, DER, EREST, FUN, IFAIL) | On-line doc: CALL GAMSDOC D04AAE (or @PRT NAG*DOC.D04AAE) | Access: LIB NBS*NAG
- D04AAF** Numerical differentiation of a function of one real variable, derivatives up to order 14. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D04AAE. | Class(es): H1 | Usage: CALL D04AAF (XVAL, NDER, HBASE, DER, EREST, FUN, IFAIL) | On-line doc: CALL GAMSDOC D04AAF (or @PRT NAG*DOC.D04AAF) | Access: LIB NBS*NAG
- D05AAE** Linear non-singular Fredholm integral equation, 2nd kind, split kernel. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D05AAF. | Class(es): I3 | Usage: CALL D05AAE (LAMBDA, A, B, K1, K2, G, F, C, N, IND, W1, W2, WD, NMAX, MN, IFAIL) | On-line doc: CALL GAMSDOC D05AAE (or @PRT NAG*DOC.D05AAE) | Access: LIB NBS*NAG
- D05AAF** Linear non-singular Fredholm integral equation, 2nd kind, split kernel. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D05AAE. | Class(es): I3 | Usage: CALL D05AAF (LAMBDA, A, B, K1, K2, G, F, C, N, IND, W1, W2, WD, NMAX, MN, IFAIL) | On-line doc: CALL GAMSDOC D05AAF (or @PRT NAG*DOC.D05AAF) | Access: LIB NBS*NAG
- D05ABE** Linear non-singular Fredholm integral equation, 2nd kind, smooth kernel. | Proprietary single precision Fortran subprogram in NAG

- library. Double precision version is D05ABF. | Class(es): I3 | Usage: CALL D05ABE (K, G, LAMBDA, A, B, ODOREV, EV, N, CM, F1, WK, NMAX, NT2P1, F, C, IFAIL) | On-line doc: CALL GAMSDOC D05ABE (or @PRT NAG*DOC.D05ABE) | Access: LIB NBS*NAG
- D05ABF** Linear non-singular Fredholm integral equation, 2nd kind, smooth kernel. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D05ABE. | Class(es): I3 | Usage: CALL D05ABF (K, G, LAMBDA, A, B, ODOREV, EV, N, CM, F1, WK, NMAX, NT2P1, F, C, IFAIL) | On-line doc: CALL GAMSDOC D05ABF (or @PRT NAG*DOC.D05ABF) | Access: LIB NBS*NAG
- D1MACH** Provides machine dependent information, e.g. D1MACH(4) returns double precision machine epsilon. | Portable double precision Fortran subprogram in MACHCONST sublibrary of CMLIB library. Single precision version is R1MACH. | Class(es): R1 | Usage: D=D1MACH(I) | On-line doc: CALL GAMSDOC D1MACH (or @PRT CMLIB*DOC.D1MACH/MACHCONST) | Tests: CMLIB*TEST-SOURCE.\$Q/MACHCONST | Access: LIB NBS*CMLIB
- D1MACH** Provides machine dependent double precision constants for the PORT library programs. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is R1MACH. | Class(es): R1 | Usage: D = D1MACH (I) | On-line doc: CALL GAMSDOC D1MACH (or @PRT PORT*DOC.D1MACH) | Access: LIB NBS*PORT
- DACOS** Arc cosine of double precision argument, $\cos^{-1}(d)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4a | Usage: D1=DACOS(D) | On-line doc: CALL GAMSDOC DACOS (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DACOSH** Arc hyperbolic cosine of double precision argument, $\cosh^{-1}(d)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is ACOSH. | Class(es): C4c | Usage: D1=DACOSH(D) | On-line doc: CALL GAMSDOC DACOSH (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DACOSH** Computes hyperbolic arccosine, $\operatorname{arccosh}$. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ACOSH. | Class(es): C4c | Usage: D = DACOSH (X) | On-line doc: CALL GAMSDOC DACOSH (or @PRT PORT*DOC.DACOSH) | Access: LIB NBS*PORT
- DAI** Airy function of double precision argument, $Ai(d)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is AI. | Class(es): C10d | Usage: D1=DAI(D) | On-line doc: CALL GAMSDOC DAI (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DAIE** Airy function of double precision argument, $-Ai(d)$, $d \leq 0$; $\exp(2/3 * d^{3/2}) * Ai(d)$, $d > 0$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is AIE. | Class(es): C10d | Usage: D1=DAIE(D) | On-line doc: CALL GAMSDOC DAIE (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DARCOS** Computes $\arccos(x)$, answer in radians. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ARCOS. | Class(es): C4a | Usage: D = DARCOS (X) | On-line doc: CALL GAMSDOC DARCOS (or @PRT PORT*DOC.DARCOS) | Access: LIB NBS*PORT
- DARSIN** Computes $\arcsin(x)$, answer in radians. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ARSIN. | Class(es): C4a | Usage: D = DARSIN (X) | On-line doc: CALL GAMSDOC DARSIN (or @PRT PORT*DOC.DARSIN) | Access: LIB NBS*PORT
- DASIN** Arc sine of double precision argument, $\sin^{-1}(d)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4a | Usage: D1=DASIN(D) | On-line doc: CALL GAMSDOC DASIN (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DASINH** Arc hyperbolic sine of double precision argument, $\sinh^{-1}(d)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is ASINH. | Class(es): C4c | Usage: D1=DASINH(D) | On-line doc: CALL GAMSDOC DASINH (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DASINH** Computes hyperbolic arcsine, $\operatorname{acrsin}(x)$. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ASINH. | Class(es): C4c | Usage: D = DASINH (X) | On-line doc: CALL GAMSDOC DASINH (or @PRT PORT*DOC.DASINH) | Access: LIB NBS*PORT
- DASUM** Compute double precision sum of absolute values of vector. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. Single precision version is SASUM. | Class(es): D1a3a | Usage: D = DASUM(N,DX,INCX) | On-line doc: CALL GAMSDOC DASUM (or @PRT CMLIB*DOC.DASUM/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- DATANH** Arc hyperbolic tangent of double precision argument, $\tanh^{-1}(d)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is ATANH. | Class(es): C4c | Usage: D1=DATANH(D) | On-line doc: CALL GAMSDOC DATANH (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DATANH** Computes hyperbolic arctangent, $\operatorname{acrtanh}(x)$. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ATANH. | Class(es): C4c | Usage: D = DATANH (X) | On-line doc: CALL GAMSDOC DATANH (or @PRT PORT*DOC.DATANH) | Access: LIB NBS*PORT
- DAWS** Dawson's function, $F(x) = e^{-(x^2)} * \int_0^x e^{t^2} dt$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DDAWS. | Class(es): C8c | Usage: R=DAWS(X) | On-line doc: CALL GAMSDOC DAWS (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DAXPY** Compute a constant times a vector plus a vector, all double precision. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. Single precision version is SAXPY. | Class(es): D1a7 | Usage: CALL DAXPY(N,DA,DX,INCX,DY,INCY) |

On-line doc: CALL GAMSDOC DAXPY (or @PRT CMLIB*DOC.DAXPY/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

DB2INK Computes parameters of a piecewise-polynomial that interpolates a given set of two-dimensional gridded data. (Double precision version of B2INK.). | Portable double precision Fortran subprogram in DTENSORBS sublibrary of CMLIB library. Single precision version is B2INK. | Class(es): E2a | Usage: CALL DB2INK(X,NX,Y,NY,FCN,LDF,KX,KY,TX,TY,BCOEF,WORK,IFLAG) | On-line doc: CALL GAMSDOC DB2INK (or @PRT CMLIB*DOC.DB2INK/DTENSORBS) | Tests: CMLIB*TEST-SOURCE.\$Q1/DTENSORBS, CMLIB*TEST-SOURCE.\$Q2/DTENSORBS | Access: LIB NBS*CMLIB | See also: DB2VAL

DB2VAL Evaluates the two-dimensional interpolating function computed by DB2INK or one of its partial derivatives. (Double precision version of B2VAL.). | Portable double precision Fortran subprogram in DTENSORBS sublibrary of CMLIB library. Single precision version is B2VAL. | Class(es): E3 | Usage: CALL DB2VAL(XVAL,YVAL,IDX,IDY,TX,TY,NX,NY,KX,KY,BCOEF,WORK) | On-line doc: CALL GAMSDOC DB2VAL (or @PRT CMLIB*DOC.DB2VAL/DTENSORBS) | Tests: CMLIB*TEST-SOURCE.\$Q1/DTENSORBS, CMLIB*TEST-SOURCE.\$Q2/DTENSORBS | Access: LIB NBS*CMLIB

DB3INK Computes parameters of a piecewise-polynomial that interpolates a given set of three-dimensional gridded data. (Double precision version of B3INK.). | Portable double precision Fortran subprogram in DTENSORBS sublibrary of CMLIB library. Single precision version is B3INK. | Class(es): E2a | Usage: CALL DB3INK(X,NX,Y,NY,FCN,LDF,KX,KY,TX,TY,BCOEF,WORK,IFLAG) | On-line doc: CALL GAMSDOC DB3INK (or @PRT CMLIB*DOC.DB3INK/DTENSORBS) | Tests: CMLIB*TEST-SOURCE.\$Q1/DTENSORBS, CMLIB*TEST-SOURCE.\$Q2/DTENSORBS | Access: LIB NBS*CMLIB | See also: DB3VAL

DB3VAL Evaluates the three-dimensional interpolating function computed by DB3INK or one of its partial derivatives. (Double precision version of B3VAL.). | Portable double precision Fortran subprogram in DTENSORBS sublibrary of CMLIB library. Single precision version is B3VAL. | Class(es): E3 | Usage: CALL DB3VAL(XVAL,YVAL,IDX,IDY,TX,TY,NX,NY,KX,KY,BCOEF,WORK) | On-line doc: CALL GAMSDOC DB3VAL (or @PRT CMLIB*DOC.DB3VAL/DTENSORBS) | Tests: CMLIB*TEST-SOURCE.\$Q1/DTENSORBS, CMLIB*TEST-SOURCE.\$Q2/DTENSORBS | Access: LIB NBS*CMLIB

DBCEVL Bicubic spline mixed partial derivative evaluator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): E3 K6 | Usage: CALL DBCEVL (X,NX,Y,NY,C,IC,XL,YL,PDS,IER) | On-line doc: CALL GAMSDOC DBCEVL (or @PRT IMSL*DOC.DBCEVL) | Access: LIB NBS*IMSL

DBCQDU Bicubic spline quadrature. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): H2b1b2 | Usage: CALL DBCQDU (F,IFD,X,NX,Y,NY,A,B,C,D,Q,WK,IER) | On-line doc: CALL GAMSDOC DBCQDU (or @PRT IMSL*DOC.DBCQDU) | Access: LIB NBS*IMSL

DBESCI Modified Bessel functions, I, of complex argument and integer order. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is BESCI. | Class(es): C10b2 | Usage: CALL DBESCI (XR, XI, NB, BR, BI) | On-line doc: CALL GAMSDOC DBESCI (or @PRT PORT*DOC.DBESCI) | Access: LIB NBS*PORT

DBESCI Bessel functions, J, of complex argument and integer order. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is BESCI. | Class(es): C10a2 | Usage: CALL DBESCI (XR, XI, NB, BR, BI) | On-line doc: CALL GAMSDOC DBESCI (or @PRT PORT*DOC.DBESCI) | Access: LIB NBS*PORT

DBES10 Bessel Function, First kind, order zero, I sub 0 (d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BES10. | Class(es): C10b1 | Usage: D1=DBES10(D) | On-line doc: CALL GAMSDOC DBES10 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

DBES11 Bessel Function, First kind, order one, I sub 1 (d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BES11. | Class(es): C10b1 | Usage: D1=DBES11(D) | On-line doc: CALL GAMSDOC DBES11 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

DBESJ0 Bessel function, first kind order zero, J sub 0 (d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BESJ0. | Class(es): C10a1 | Usage: D1=DBESJ0(D) | On-line doc: CALL GAMSDOC DBESJ0 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

DBESJ1 Bessel function, first kind order one, J sub one (d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BESJ1. | Class(es): C10a1 | Usage: D1=DBESJ1(D) | On-line doc: CALL GAMSDOC DBESJ1 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

DBESK0 Bessel function, modified, third kind order zero, K sub 0 (d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BESK0. | Class(es): C10b1 | Usage: D1=DBESK0(D) | On-line doc: CALL GAMSDOC DBESK0 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

DBESK1 Bessel function, modified, third kind order one, K sub 1 (d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BESK1. | Class(es): C10b1 | Usage: D1=DBESK1(D) | On-line doc: CALL GAMSDOC DBESK1 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

DBESKS Compute sequence of Bessel functions, modified, of third kind, K sub v+i (d), for n values of i going up or down from zero. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BESKS. | Class(es): C10b3 | Usage: CALL DBESKS(DXNU,D,N,DBK) | On-line doc: CALL GAMSDOC DBESKS (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

DBESRI Modified Bessel functions, I, of real argument and integer order. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is BESRI. | Class(es): C10b1 | Usage: CALL DBESRI (X, NB, B) | On-line doc: CALL GAMSDOC DBESRI (or @PRT PORT*DOC.DBESRI) | Access: LIB NBS*PORT

- DBESRJ** Bessel functions, J , of real argument and integer order. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is BESRJ. | Class(es): C10a1 | Usage: CALL DBESRJ (X, NB, B) | On-line doc: CALL GAMSDOC DBESRJ (or @PRT PORT*DOC.DBESRJ) | Access: LIB NBS*PORT
- DBESY0** Bessel function, second kind order zero, $Y_{\text{sub } 0}$ (d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BESY0. | Class(es): C10a1 | Usage: D1=DBESY0(D) | On-line doc: CALL GAMSDOC DBESY0 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DBESY1** Bessel function, second kind order one, $Y_{\text{sub } 1}$ (d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BESY1. | Class(es): C10a1 | Usage: D1=DBESY1(D) | On-line doc: CALL GAMSDOC DBESY1 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DBETA** Beta function of double precision arguments, $B(a,b) = (\Gamma(a) * \Gamma(b)) / \Gamma(a+b)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BETA. | Class(es): C7b | Usage: D1=DBETA(DA,DB) | On-line doc: CALL GAMSDOC DBETA (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DBETAI** Incomplete Beta function of double precision arguments, $I_{\text{sub } x}(a,b) = B_{\text{sub } x}(a,b) / B(a,b)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BETAI. | Class(es): C7f | Usage: D=DBETAI(DX,DA,DB) | On-line doc: CALL GAMSDOC DBETAI (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DBFQAD** Integrates function times derivative of B-spline from X_1 to X_2 . The B-spline is in "B" representation. | Portable double precision Fortran subprogram in DBSPLINE sublibrary of CMLIB library. Single precision version is BFQAD. | Class(es): H2a2a1 E3 K6 | Usage: CALL DBFQAD(F,T,BCOEF,N,K,ID,X1,X2,TOL,QUAD,IERR,WORK) | On-line doc: CALL GAMSDOC DBFQAD (or @PRT CMLIB*DOC.DBFQAD/DBSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/DBSPLINE | Access: LIB NBS*CMLIB
- DBI** Double precision Bairy function of a real argument, $Bi(d1)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BI. | Class(es): C10d | Usage: D = DBI (D1) | On-line doc: CALL GAMSDOC DBI (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DBIE** Exponentially scaled double precision Bairy function, $-Bi(d1)$ for $d1 < -0$, $= \exp(-2/3 * d1^{**}(3/2)) * Bi(d1)$ for $d1 > -0$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BIE. | Class(es): C10d | Usage: D = DBIE (D1) | On-line doc: CALL GAMSDOC DBIE (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DBINOM** Double precision Binomial function, $m! / (m! * (n-m)!)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BINOM. | Class(es): C1 | Usage: D = DBINOM (N,M) | On-line doc: CALL GAMSDOC DBINOM (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DBINT4** Computes B-spline which interpolates given X,Y data with various end conditions. The "B" representation is used. | Portable double precision Fortran subprogram in DBSPLINE sublibrary of CMLIB library. Single precision version is BINT4. | Class(es): E1a | Usage: CALL DBINT4(X,Y,NDATA,IBCL,IBCR,FBCL,FCR,KNTOPT,T,BCOEF,N,K, W) | On-line doc: CALL GAMSDOC DBINT4 (or @PRT CMLIB*DOC.DBINT4/DBSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/DBSPLINE | Access: LIB NBS*CMLIB | See also: DBVALU for evaluation. See package documentation for other facilities.
- DBINTK** Produces B-spline coefficients of k-th order B-spline with given knots and with values at given points. | Portable double precision Fortran subprogram in DBSPLINE sublibrary of CMLIB library. Single precision version is BINTK. | Class(es): E1a | Usage: CALL DBINTK(X,Y,T,N,K,BCOEF,Q,WORK) | On-line doc: CALL GAMSDOC DBINTK (or @PRT CMLIB*DOC.DBINTK/DBSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/DBSPLINE | Access: LIB NBS*CMLIB | See also: DBVALU for evaluation. See package documentation for other facilities.
- DBLIN** Numerical integration of a function of two variables. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): H2b2a1 | Usage: R = DBLIN(F,AX,BX,AY,BY,AERR,ERROR,IERR) | On-line doc: CALL GAMSDOC DBLIN (or @PRT IMSL*DOC.DBLIN) | Access: LIB NBS*IMSL
- DBQUAD** Adaptively integrates functions which have discontinuities in their derivatives. User can specify these points. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is BQUAD. | Class(es): H2a2a1 | Usage: CALL DBQUAD (F,N,X,EPS,ANS,ERREST) | On-line doc: CALL GAMSDOC DBQUAD (or @PRT PORT*DOC.DBQUAD) | Access: LIB NBS*PORT
- DBS10E** Modified (hyperbolic) Bessel function, dble precision, of special integer order scaled by an exponential, First kind, order zero, $e^{**}d1 / I_{\text{sub } 0}(d1)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BES10E. | Class(es): C10b1 | Usage: D = DBS10E (D1) | On-line doc: CALL GAMSDOC DBS10E (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DBS11E** Modified (hyperbolic) Bessel function, dble precision, of special integer order scaled by an exponential, First kind, order one, $e^{**}d1 / I_{\text{sub } 1}(d1)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BES11E. | Class(es): C10b1 | Usage: D = DBS11E (D1) | On-line doc: CALL GAMSDOC DBS11E (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DBSK0E** Modified (hyperbolic) Bessel function, dble precision, of special integer order scaled by an exponential, Third kind, order zero, $e^{**}d1 * K_{\text{sub } 0}(d1)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BESK0E. | Class(es): C10b1 | Usage: D = DBSK0E (D1) | On-line doc: CALL GAMSDOC DBSK0E (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DBSK1E** Modified (hyperbolic) Bessel function, dble. precision, of special integer orderscaled by an exponential, Third kind, order one, $e^{**}d1 *$

- K sub 1 (d1).** | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BESK1E. | Class(es): C10b1 | Usage: D = DBSK1E (D1) | On-line doc: CALL GAMSDOC DBSK1E (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DBSKES** Sequences of Bessel fns. scaled by an exponential. Modified 3rd kind. $e^{**x} * K \text{ sub } v+i (d), /n/$ vals. comptd. where $i=0,1,...n-1$ for $n>0$, $i=0,-1,...n+1$ for $n<0$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BESKES. | Class(es): C10b3 | Usage: CALL DBSKES(DXNU,D,N,DBK) | On-line doc: CALL GAMSDOC DBSKES (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DBSPDR** Construct divided difference table from "B" representation of B-spline for a derivative calculation. | Portable double precision Fortran subprogram in DBSPLINE sublibrary of CMLIB library. Single precision version is BSPDR. | Class(es): E3 K6 | Usage: CALL DBSPDR(T,A,N,K,NDERIV,AD) | On-line doc: CALL GAMSDOC DBSPDR (or @PRT CMLIB*DOC.DBSPDR/DBSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/DBSPLINE | Access: LIB NBS*CMLIB
- DBSPEV** Calculates the value of the spline and its derivatives at X from its "B" representation. | Portable double precision Fortran subprogram in DBSPLINE sublibrary of CMLIB library. Single precision version is BSPEV. | Class(es): E3 K6 | Usage: CALL DBSPEV(T,AD,N,K,NDERIV,X,INEV,SVALUE,WORK) | On-line doc: CALL GAMSDOC DBSPEV (or @PRT CMLIB*DOC.DBSPEV/DBSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/DBSPLINE | Access: LIB NBS*CMLIB
- DBSPL1** Evaluates at a given set of points in a specified mesh interval, basis splines together with selected orders of derivatives. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is BSPL1. | Class(es): E3 K6 | Usage: CALL DBSPL1(K,T,N,X,NX,ILEFT,ID,NID,BX) | On-line doc: CALL GAMSDOC DBSPL1 (or @PRT PORT*DOC.DBSPL1) | Access: LIB NBS*PORT
- DBSPLI** Obtains the integrals of basis splines, from the left-most mesh point to a specified set of points. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is BSPLI. | Class(es): H2a2a1 E3 K6 | Usage: CALL DBSPLI(K,T,N,X,NX,ILEFT,BIX) | On-line doc: CALL GAMSDOC DBSPLI (or @PRT PORT*DOC.DBSPLI) | Access: LIB NBS*PORT
- DBSPLN** Evaluates at a given set of points in a specified mesh interval, all the basis splines which are nonzero in that interval. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is BSPLN. | Class(es): E3 K6 | Usage: CALL DBSPLN(K,T,N,X,NX,ILEFT,BX) | On-line doc: CALL GAMSDOC DBSPLN (or @PRT PORT*DOC.DBSPLN) | Access: LIB NBS*PORT
- DBSPPP** Converts from "B" representation of B-spline to piecewise polynomial representation. | Portable double precision Fortran subprogram in DBSPLINE sublibrary of CMLIB library. Single precision version is BSPPP. | Class(es): E3 K6 | Usage: CALL DBSPPP(T,A,N,K,LDC,C,X1,LX1,WORK) | On-line doc: CALL GAMSDOC DBSPPP (or @PRT CMLIB*DOC.DBSPPP/DBSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/DBSPLINE | Access: LIB NBS*CMLIB
- DBSPVD** Calculates value and derivatives of order less than NDERIV of all B-spline basis functions which do not vanish at X. | Portable double precision Fortran subprogram in DBSPLINE sublibrary of CMLIB library. Single precision version is BSPVD. | Class(es): E3 K6 | Usage: CALL DBSPVD(T,K,NDERIV,X,ILEFT,LDVNIK,VNIKX,WORK) | On-line doc: CALL GAMSDOC DBSPVD (or @PRT CMLIB*DOC.DBSPVD/DBSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/DBSPLINE | Access: LIB NBS*CMLIB
- DBSPVN** Calculates the value of all (possibly) nonzero B-spline basis functions at X of a given order. | Portable double precision Fortran subprogram in DBSPLINE sublibrary of CMLIB library. Single precision version is BSPVN. | Class(es): E3 K6 | Usage: CALL DBSPVN(T,JHIGH,K,INDEX,X,ILEFT,VNIKX,WORK,IWORK) | On-line doc: CALL GAMSDOC DBSPVN (or @PRT CMLIB*DOC.DBSPVN/DBSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/DBSPLINE | Access: LIB NBS*CMLIB
- DBSQAD** Computes the integral of a B-spline from X1 to X2. The B-spline must be in "B" representation. | Portable double precision Fortran subprogram in DBSPLINE sublibrary of CMLIB library. Single precision version is BSQAD. | Class(es): H2a2a1 E3 K6 | Usage: CALL DBSQAD(T,BCOEF,N,K,X1,X2,BQUAD,WORK) | On-line doc: CALL GAMSDOC DBSQAD (or @PRT CMLIB*DOC.DBSQAD/DBSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/DBSPLINE | Access: LIB NBS*CMLIB
- DBURAM** Finds the best uniform rational approximation to a given function on a specified mesh. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is BURAM. | Class(es): K2 | Usage: CALL DBURAM (NPTS, MESH, FN, M, N, P, Q, DELK) | On-line doc: CALL GAMSDOC DBURAM (or @PRT PORT*DOC.DBURAM) | Access: LIB NBS*PORT | See also: DTCHBP
- DBURM1** Finds best uniform rational approximation to a given function on a specified mesh, starting from a given initial approximation. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is BURM1. | Class(es): K2 | Usage: CALL DBURM1 (NPTS, MESH, FN, MAXITR, ITOL, M, N, P, Q, DELK) | On-line doc: CALL GAMSDOC DBURM1 (or @PRT PORT*DOC.DBURM1) | Access: LIB NBS*PORT | See also: DTCHBP
- DBVALU** Calculates (at X) the value of the IDERIV-th derivative of the B-spline from its "B" representation. | Portable double precision Fortran subprogram in DBSPLINE sublibrary of CMLIB library. Single precision version is BVALU. | Class(es): E3 K6 | Usage: CALL DBVALU(T,A,N,K,IDERIV,X,INBV,WORK) | On-line doc: CALL GAMSDOC DBVALU (or @PRT CMLIB*DOC.DBVALU/DBSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/DBSPLINE | Access: LIB NBS*CMLIB
- DCADRE** Numerical integration of a function using cautious adaptive Romberg extrapolation. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): H2a1a1 | Usage: X = DCADRE (F,A,B,AERR,RERR,ERROR,IER) | On-line doc: CALL

GAMSDOC DCADRE (or @PRT IMSL*DOC.DCADRE) | Access: LIB NBS*IMSL

DCBRT Double precision cube root of a double precision argument. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is CBRT. | Class(es): C2 | Usage: D = DCBRT (D1) | On-line doc: CALL GAMSDOC DCBRT (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

DCDOT Computes a complex precision dot product using double precision accumulation. | Portable single precision Fortran subprogram in XBLAS sublibrary of CMLIB library. | Class(es): D1a4 | Usage: CALL DCDOT(N,FM,CX,INCX,CY,INCY,DCR,DCI) | On-line doc: CALL GAMSDOC DCDOT (or @PRT CMLIB*DOC.DCDOT/XBLAS) | Access: LIB NBS*CMLIB

DCEIL Finds the smallest integer greater than or equal to x. Input and output are double precision. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is CEIL. | Class(es): C1 | Usage: D = DCEIL (X) | On-line doc: CALL GAMSDOC DCEIL (or @PRT PORT*DOC.DCEIL) | Access: LIB NBS*PORT

DCHDC Compute Cholesky decomposition of positive definite double precision matrix with optional pivoting. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SCHDC. | Class(es): D2b1b | Usage: CALL DCHDC(A,LDA,P,WORK,JPVT,JOB,INFO) | On-line doc: CALL GAMSDOC DCHDC (or @PRT CMLIB*DOC.DCHDC/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB

DCHDD Downdates Cholesky factorization of positive definite double precision matrix. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SCHDD. | Class(es): D7b | Usage: CALL DCHDD(R,LDR,P,X,Z,LDZ,NZ,Y,RHO,C,S,INFO) | On-line doc: CALL GAMSDOC DCHDD (or @PRT CMLIB*DOC.DCHDD/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DCHDC

DCHEX Updates Cholesky factorization of positive definite double precision matrix. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SCHEX. | Class(es): D7b | Usage: CALL DCHEX(R,LDR,P,K,L,Z,LDZ,NZ,C,S,JOB) | On-line doc: CALL GAMSDOC DCHEX (or @PRT CMLIB*DOC.DCHEX/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DCHDC

DCHU Double precision confluent hypergeometric, U(da,db,d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is CHU. | Class(es): C11 | Usage: D1 = DCHU (DA,DB,D) | On-line doc: CALL GAMSDOC DCHU (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

DCHUD Updates Cholesky factorization of positive definite double precision matrix. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SCHUD. | Class(es): D7b | Usage: CALL DCHUD(R,LDR,P,X,Z,LDZ,NZ,Y,RHO,C,S) | On-line doc: CALL GAMSDOC DCHUD (or @PRT CMLIB*DOC.DCHUD/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DCHDC

DCLINQ Solves a complex system of linear equations. Coefficient matrix must be input as two real matrices. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is CLINQ. | Class(es): D2c1 | Usage: CALL DCLINQ (N,AR,AI,BR,BI,NB,XR,XI) | On-line doc: CALL GAMSDOC DCLINQ (or @PRT PORT*DOC.DCLINQ) | Access: LIB NBS*PORT

DCLST2 Finds the least squares solution of a complex linear algebraic system (double precision) of equations $AX=B$. B may be a matrix. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is CLST2. | Class(es): D9 | Usage: CALL DCLST2 (MDIM,NDIM,M,N,AR,AI,BR,BI,NB,XR,XI) | On-line doc: CALL GAMSDOC DCLST2 (or @PRT PORT*DOC.DCLST2) | Access: LIB NBS*PORT

DCOPY Copy a vector X to a vector Y, both double precision. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. Single precision version is SCOPY. | Class(es): D1a5 | Usage: CALL DCOPY(N,DX,INCX,DY,INCY) | On-line doc: CALL GAMSDOC DCOPY (or @PRT CMLIB*DOC.DCOPY/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

DCOSDG Double precision cosine d1 in degrees, cos(d1). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is COSDG. | Class(es): C4a | Usage: D = DCOSDG (D1) | On-line doc: CALL GAMSDOC DCOSDG (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

DCOSH Computes hyperbolic cosine, cosh(x). | Proprietary double precision Fortran subprogram in PORT library. Single precision version is COSH. | Class(es): C4c | Usage: D = DCOSH (X) | On-line doc: CALL GAMSDOC DCOSH (or @PRT PORT*DOC.DCOSH) | Access: LIB NBS*PORT

DCOT Double precision cotangent, cot(d1). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is COT. | Class(es): C4a | Usage: D = DCOT (D1) | On-line doc: CALL GAMSDOC DCOT (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

DCPOLY Finds zeros of polynomial with double precision complex coefficients. Uses real double precision arrays to represent complex numbers. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is CPOLY. | Class(es): F1a1b | Usage: CALL DCPOLY (DEGREE,OPR,OPI,ZEROR,ZEROI) | On-line doc: CALL GAMSDOC DCPOLY (or @PRT PORT*DOC.DCPOLY) | Access: LIB NBS*PORT

DCSEVL Evaluates an n term series of Chebyshev polynomials at a given point. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is CSEVL. | Class(es): C3a2 | Usage: D=DCSEVL(DX,DCS,N) | On-line doc: CALL GAMSDOC DCSEVL (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB | See also: INITDS

DCSEVU Cubic spline first and second derivative evaluator. | Proprietary single precision Fortran subprogram in IMSL library. |

- Class(es):** E3 K6 | **Usage:** CALL DCSEVU (X,Y,NX,C,IC,U,DS,M1,DDS,M2,IER) | **On-line doc:** CALL GAMSDOC DCSEVU (or @PRT IMSL*DOC.DCSEVU) | **Access:** LIB NBS*IMSL
- DCSPD1** Finds a numerical approximation to the first derivative at requested points in given input data by using spline interpolation. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is CSPD1. | **Class(es):** H1 | **Usage:** CALL DCSPD1 (X,Y,N,XX,YY,YYP,NN) | **On-line doc:** CALL GAMSDOC DCSPD1 (or @PRT PORT*DOC.DCSPD1) | **Access:** LIB NBS*PORT
- DCSPFE** Evaluates a cubic spline function which has already been fit to n data pairs (x, y) by DCSPF1. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is CSPFE. | **Class(es):** E3 K6 | **Usage:** CALL DCSPFE (X,Y,YP,YPP,N,XX,YY,NN) | **On-line doc:** CALL GAMSDOC DCSPFE (or @PRT PORT*DOC.DCSPFE) | **Access:** LIB NBS*PORT
- DCSPF1** Fits a cubic spline function to n input data pairs (x,y). Produces and interpolatory, not least squares fit. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is CSPF1. | **Class(es):** E1a | **Usage:** CALL DCSPF1 (X,Y,N,B,YP,YPP) | **On-line doc:** CALL GAMSDOC DCSPF1 (or @PRT PORT*DOC.DCSPF1) | **Access:** LIB NBS*PORT | **See also:** DCSPFE
- DCSPIN** Interpolates at requested points in given input data using a spline approximation. Interpolation, not a least squares fit. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is CSPIN. | **Class(es):** E1a | **Usage:** CALL DCSPIN (X,Y,N,XX,YY,NN) | **On-line doc:** CALL GAMSDOC DCSPIN (or @PRT PORT*DOC.DCSPIN) | **Access:** LIB NBS*PORT
- DCSPQU** Finds the integral of a function defined by pairs (x,y) of input points. The x's can be unequally spaced. Uses spline interpolation. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is CSPQU. | **Class(es):** H2a1b2 | **Usage:** CALL DCSPQU (X,Y,N,XLOW,XHIGH,ANS) | **On-line doc:** CALL GAMSDOC DCSPQU (or @PRT PORT*DOC.DCSPQU) | **Access:** LIB NBS*PORT
- DCSQDU** Cubic spline quadrature. | Proprietary single precision Fortran subprogram in IMSL library. | **Class(es):** H2a2a1 E3 K6 | **Usage:** CALL DCSQDU (X,Y,NX,C,IC,A,B,Q,IER) | **On-line doc:** CALL GAMSDOC DCSQDU (or @PRT IMSL*DOC.DCSQDU) | **Access:** LIB NBS*IMSL
- DDASSL** Solves the system of differential/algebraic equations of the form $g(t, y, y_{\text{prime}}) = 0$, with given initial values. | Portable double precision Fortran subprogram in DDASSL sublibrary of CMLIB library. Single precision version is SDASSL. | **Class(es):** 11a1b | **Usage:** CALL DDASSL(RES,NEQ,T,Y,YPRIME,TOUT,INFO,RTOL,ATOL,LDID,RWORK,LRW,IWORK,LIW,RPAR,IPAR,JAC) | **On-line doc:** CALL GAMSDOC DDASSL (or @PRT CMLIB*DOC.DDASSL/SDASSL) | **Tests:** CMLIB*TEST-SOURCE.DDASSL/SDASSL | **Access:** LIB NBS*CMLIB
- DDAWS** Double precision Dawson function $= e^{**(-d1**2)} * \text{the integral from } 0 \text{ to } d1 \text{ of } e^{**(-t**2)} dt$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is DAWs. | **Class(es):** C8c | **Usage:** D = DDAWS (D1) | **On-line doc:** CALL GAMSDOC DDAWS (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | **Access:** LIB NBS*CMLIB
- DDL2SF** Fits discrete data with a B-spline of order K, by least squares. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is DL2SF. | **Class(es):** K1a1a1 | **Usage:** CALL DDL2SF (X,Y,NXY,K,T,NT,A) | **On-line doc:** CALL GAMSDOC DDL2SF (or @PRT PORT*DOC.DDL2SF) | **Access:** LIB NBS*PORT | **See also:** DSPLNE DSPLND DSPLNI DSPLN1 DSPLN2
- DDL2SW** Fits discrete data with a B-spline of order k, by weighted least squares. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is DL2SW. | **Class(es):** K1a1a1 | **Usage:** CALL DDL2SW (X,Y,NXY,W,K,T,NT,A) | **On-line doc:** CALL GAMSDOC DDL2SW (or @PRT PORT*DOC.DDL2SW) | **Access:** LIB NBS*PORT | **See also:** DSPLNE DSPLND DSPLNI DSPLN1 DSPLN2
- DDOT** Compute double precision dot product. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. Single precision version is SDOT. | **Class(es):** D1a4 | **Usage:** CALL DDOT(N,DX,INCX,DY,INCY) | **On-line doc:** CALL GAMSDOC DDOT (or @PRT CMLIB*DOC.DDOT/BLAS) | **Tests:** CMLIB*TEST-SOURCE.\$Q/BLAS | **Access:** LIB NBS*CMLIB
- DDRIV1** Numerical Integration, Initial Value Problems, Ordinary Differential Eqs., Gear Stiff Formulas, Easy to Use. | Portable double precision Fortran subprogram in DDRIV sublibrary of CMLIB library. Single precision version is SDRIV1. | **Class(es):** 11a2 11a1b | **Usage:** CALL DDRIV1(N,T,Y,TOUT,MSTATE,EPS,WORK,LENW) | **On-line doc:** CALL GAMSDOC DDRIV1 (or @PRT CMLIB*DOC.SUMMARY/DDRIV) | **Tests:** CMLIB*TEST-SOURCE.\$F/DDRIV, CMLIB*TEST-SOURCE.\$Q/DDRIV | **Access:** LIB NBS*CMLIB
- DDRIV2** Numerical Integration, Initial Value Problems, Ordinary Differential Eqs., Gear / Adams Formulas. | Portable double precision Fortran subprogram in DDRIV sublibrary of CMLIB library. Single precision version is SDRIV2. | **Class(es):** 11a2 11a1b | **Usage:** CALL DDRIV2(N,T,Y,F,TOUT,MSTATE,IROOT,EPS,EWT,MINT,WORK,LENW,IWORK,LENIW,G) | **On-line doc:** CALL GAMSDOC DDRIV2 (or @PRT CMLIB*DOC.SUMMARY/DDRIV) | **Tests:** CMLIB*TEST-SOURCE.\$F/DDRIV, CMLIB*TEST-SOURCE.\$Q/DDRIV | **Access:** LIB NBS*CMLIB
- DDRIV3** Numerical Integration, Initial Value Problem, Ordinary Differential Eqs., Implicit Eqs., Sparse Jacobians. | Portable double precision Fortran subprogram in DDRIV sublibrary of CMLIB library. Single precision version is SDRIV3. | **Class(es):** 11a2 11a1b | **Usage:** CALL DDRIV3(N,T,Y,F,NSTATE,TOUT,NTASK,IROOT,EPS,EWT,IERROR,MINT,MITER,IMPL,ML,MU,MXORD,HMAX,WORK,LENW,IWORK,LENIW,JACOBN,FA,NDE,MXSTEP,G) | **On-line doc:** CALL GAMSDOC DDRIV3 (or @PRT CMLIB*DOC.SUMMARY/DDRIV) | **Tests:** CMLIB*TEST-SOURCE.\$F/DDRIV, CMLIB*TEST-SOURCE.\$Q/DDRIV | **Access:** LIB NBS*CMLIB
- DE1** Double precision exponential integral, the integral from d1 to infinity of $(e^{**(-t)} / t) dt$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is E1. | **Class(es):** C5 | **Usage:** D = DE1 (D1) | **On-line doc:** CALL GAMSDOC DE1 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | **Access:** LIB NBS*CMLIB
- DEABM** Solves a system of first order ordinary differential equations with arbitrary initial conditions by a predictor-corrector method. | Portable single precision Fortran subprogram in DEPAC sublibrary of CMLIB library. | **Class(es):** 11a1b | **Usage:** CALL

- DEABM(F,NEQ,T,Y,TOUT,INFO,RTOL,ATOL,IDID,RWORK,LRW, IWORK, LIW,RPAR,IPAR) | On-line doc: CALL GAMSDOC DEABM (or @PRT CMLIB*DOC.DEABM/DEPAC) | Tests: CMLIB*TEST-SOURCE.DEABM/DEPAC | Access: LIB NBS*CMLIB
- DEBDF Solves a system of first order stiff ordinary differential equations with arbitrary initial conditions by Gear's method. | Portable single precision Fortran subprogram in DEPAC sublibrary of CMLIB library. | Class(es): 11a2 | Usage: CALL DEBDF(F,NEQ,T,Y,TOUT,INFO,RTOL,ATOL,IDID,RWORK,LRW, IWORK, LIW,RPAR,IPAR,JAC) | On-line doc: CALL GAMSDOC DEBDF (or @PRT CMLIB*DOC.DEBDF/DEPAC) | Tests: CMLIB*TEST-SOURCE.DEBDF/DEPAC | Access: LIB NBS*CMLIB
- DEEBSF Estimates the error in a given B-spline fit to a function f by refining the mesh intervals selected by user. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is EEBF. | Class(es): E3 K6 | Usage: D = DEEBSF (K,T1,N1,A1,T2,N2,A2) | On-line doc: CALL GAMSDOC DEEBSF (or @PRT PORT*DOC.DEEBSF) | Access: LIB NBS*PORT
- DEEBSI Estimates the error in a given B-spline fit to a function f by refining the mesh intervals selected by user. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is EEBI. | Class(es): E3 K6 | Usage: X = DEEBSI (K,T1,N1,A1,T2,N2,A2,X,NX,EREST1,EREST2) | On-line doc: CALL GAMSDOC DEEBSI (or @PRT PORT*DOC.DEEBSI) | Access: LIB NBS*PORT
- DEESFF Finds the maximum absolute error in a given B-spline fit to a function, f . | Proprietary double precision Fortran subprogram in PORT library. Single precision version is EESFF. | Class(es): E3 K6 | Usage: D = DEESFF (K,T,N,A,F) | On-line doc: CALL GAMSDOC DEESFF (or @PRT PORT*DOC.DEESFF) | Access: LIB NBS*PORT
- DEESFI Finds the maximum absolute error in a given B-spline fit to a function, f , on a set of user selected intervals. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is EESFI. | Class(es): E3 K6 | Usage: D = DEESFI (K,T,N,A,F,X,NX,EEST) | On-line doc: CALL GAMSDOC DEESFI (or @PRT PORT*DOC.DEESFI) | Access: LIB NBS*PORT
- DEFINE Defines a vector of constants by setting all of the elements in the single precision vector X equal to $XNEW$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): D1a1 | Usage: CALL DEFINE(X,N,XNEW) | On-line doc: CALL GAMSDOC DEFINE (or @PRT DATAPAC*DOC.DEFINE) | Access: LIB NBS*DATAPAC
- DEI Double precision exponential integral, the integral from $-d1$ to infinity of $(e^{-t} - t) / t$ dt. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is EI. | Class(es): C5 | Usage: D = DEI (D1) | On-line doc: CALL GAMSDOC DEI (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DEIGEN Finds all eigenvalues and eigenvectors of a real matrix. Output consists of pairs of real arrays. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is EIGEN. | Class(es): D4a2 | Usage: CALL DEIGEN (NM,N,A,WR,WI,Z) | On-line doc: CALL GAMSDOC DEIGEN (or @PRT PORT*DOC.DEIGEN) | Access: LIB NBS*PORT
- DELETE Deletes all observations in the vector X which are inside the interval $[XMIN, XMAX]$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L2d | Usage: CALL DELETE(X,N,XMIN,XMAX,NEWN) | On-line doc: CALL GAMSDOC DELETE (or @PRT DATAPAC*DOC.DELETE) | Access: LIB NBS*DATAPAC
- DEMODO Performs a complex demodulation on the data in the input vector X at the input demodulation frequency = F . | Class(es): L10d | Usage: CALL DEMODO(X,N,F) | On-line doc: CALL GAMSDOC DEMODO (or @PRT DATAPAC*DOC.DEMODO) | Access: LIB NBS*DATAPAC
- DERF Double precision error function, $(2 / \sqrt{\pi}) * \int_0^{d1} e^{-t^2} dt$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is ERF. | Class(es): C8a L5a1e | Usage: D = DERF (D1) | On-line doc: CALL GAMSDOC DERF (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DERFC Double precision complementary error function, $(2 / \sqrt{\pi}) * \int_{d1}^{\infty} e^{-t^2} dt$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is ERFC. | Class(es): C8a L5a1e | Usage: D = DERFC (D1) | On-line doc: CALL GAMSDOC DERFC (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DERKF Solves a system of first order ordinary differential equations with arbitrary initial conditions by a Runge-Kutta method. | Portable single precision Fortran subprogram in DEPAC sublibrary of CMLIB library. | Class(es): 11a1a | Usage: CALL DERKF(F,NEQ,T,Y,TOUT,INFO,RTOL,ATOL,IDID,RWORK,LRW, IWORK, LIW,RPAR,IPAR) | On-line doc: CALL GAMSDOC DERKF (or @PRT CMLIB*DOC.DERKF/DEPAC) | Tests: CMLIB*TEST-SOURCE.DERKF/DEPAC | Access: LIB NBS*CMLIB
- DEXCDF Computes the cumulative distribution function value for the double exponential (Laplace) distribution with mean = 0. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1d | Usage: CALL DEXCDF(X,CDF) | On-line doc: CALL GAMSDOC DEXCDF (or @PRT DATAPAC*DOC.DEXCDF) | Access: LIB NBS*DATAPAC
- DEXINT Computes sequences of exponential integrals $E(N+K,X)$ $K=0, \dots, M-1$ or $\exp(X)$ times same to specified tolerance. | Portable double precision Fortran subprogram in AMOSLIB sublibrary of CMLIB library. Single precision version is EXINT. | Class(es): C5 | Usage: CALL DEXINT(X,N,KODE,TOL,EN,IERR) | On-line doc: CALL GAMSDOC DEXINT (or @PRT CMLIB*DOC.DEXINT/AMOSLIB) | Tests: CMLIB*TEST-SOURCE.DEXINT/AMOSLIB | Access: LIB NBS*CMLIB
- DEXPDF Computes the probability density function value for the double exponential (Laplace) distribution with mean = 0. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1d | Usage: CALL DEXPDF(X,PDF) | On-line doc: CALL GAMSDOC DEXPDF (or @PRT DATAPAC*DOC.DEXPDF) | Access: LIB NBS*DATAPAC
- DEXPLT Generates a double exponential (Laplace) probability plot with mean = 0 and standard deviation = $\sqrt{2}$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c4d | Usage: CALL DEXPLT(X,N) | On-line doc: CALL GAMSDOC DEXPLT (or @PRT DATAPAC*DOC.DEXPLT) | Access: LIB NBS*DATAPAC

- DEXPPF** Computes the percent point function value for the double exponential (Laplace) distribution with mean = 0. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2d | Usage: CALL DEXPPF(P,PPF) | On-line doc: CALL GAMSDOC DEXPPF (or @PRT DATAPAC*DOC.DEXPPF) | Access: LIB NBS*DATAPAC
- DEXPRL** Double precision relative error exponential from first order, $((e^{**d1}) - 1) / x$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is EXPREL. | Class(es): C4b | Usage: D = DEXPRL (D1) | On-line doc: CALL GAMSDOC DEXPRL (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DEXRAN** Generates a random sample of size N from the double exponential (Laplace) distribution with mean = 0 and standard deviation = sqrt(2). | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a4 | Usage: CALL DEXRAN(N,ISTART,X) | On-line doc: CALL GAMSDOC DEXRAN (or @PRT DATAPAC*DOC.DEXRAN) | Access: LIB NBS*DATAPAC
- DEXSF** Computes the sparsity function value for the double exponential (Laplace) distribution with mean = 0 and standard deviation = sqrt(2). | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2d | Usage: CALL DEXSF(P,SF) | On-line doc: CALL GAMSDOC DEXSF (or @PRT DATAPAC*DOC.DEXSF) | Access: LIB NBS*DATAPAC
- DFAC** Double precision factorial, nl. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is FAC. | Class(es): C1 | Usage: D = DFAC (N) | On-line doc: CALL GAMSDOC DFAC (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DFFT** Compute FFT of complex data sequence (forward or reverse) any number of points. Useful for multivariate transforms. Uses no complex arithmetic. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is FFT. | Class(es): J1a2 J1b | Usage: CALL DFFT (A,B,NTOT,N,NSPAN,ISN) | On-line doc: CALL GAMSDOC DFFT (or @PRT PORT*DOC.DFFT) | Access: LIB NBS*PORT
- DFFTC** Mixed radix fast Fourier transform of complex data. Two arrays used for complex data. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is FFTC. | Class(es): J1a2 | Usage: CALL DFFTC (N,AR,A1) | On-line doc: CALL GAMSDOC DFFTC (or @PRT PORT*DOC.DFFTC) | Access: LIB NBS*PORT | See also: DFFTC1
- DFFTC1** Finds the inverse fast Fourier transform, given the Fourier coefficients in the frequency domain. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is FFTC1. | Class(es): J1a2 | Usage: CALL DFFTC1 (N,FR,F1) | On-line doc: CALL GAMSDOC DFFTC1 (or @PRT PORT*DOC.DFFTC1) | Access: LIB NBS*PORT | See also: DFFTC
- DFFTR** Mixed radix fast Fourier transform to find the transform of 2N real data points. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is FFTR. | Class(es): J1a1 | Usage: CALL DFFTR (NNP2,A,B) | On-line doc: CALL GAMSDOC DFFTR (or @PRT PORT*DOC.DFFTR) | Access: LIB NBS*PORT | See also: DFFTR1
- DFFTR1** Finds the inverse Fourier transform using Fourier coefficients assumed to arise from real data in the time domain. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is FFTR1. | Class(es): J1a1 | Usage: CALL DFFTR1 (NN,FR,F1) | On-line doc: CALL GAMSDOC DFFTR1 (or @PRT PORT*DOC.DFFTR1) | Access: LIB NBS*PORT | See also: DFFTR
- DFLR** Finds the largest integer less than or equal to x. Input and output are double precision. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is FLR. | Class(es): C1 | Usage: D = DFLR (X) | On-line doc: CALL GAMSDOC DFLR (or @PRT PORT*DOC.DFLR) | Access: LIB NBS*PORT
- DFMIN** Finds an approximate local minimum of a univariate user defined EXTERNAL function, f. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is FMIN. | Class(es): G1a2 | Usage: D = DFMIN (F,X,A,B,T) | On-line doc: CALL GAMSDOC DFMIN (or @PRT PORT*DOC.DFMIN) | Access: LIB NBS*PORT
- DGAMI** Double precision incomplete gamma fn., the integral from 0 to d of $(t^{**}(da-1) * e^{**t}) dt$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is GAM1. | Class(es): C7e | Usage: D1 = DGAMI (DA,D) | On-line doc: CALL GAMSDOC DGAMI (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DGAMIC** Double precision complementary incomplete gamma function, the integral from d to infinity of $(t^{**}(da-1) * e^{**t}) dt$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is GAMIC. | Class(es): C7e | Usage: D1 = DGAMIC (DA,D) | On-line doc: CALL GAMSDOC DGAMIC (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DGAMIT** Double precision Triconi's incomplete gamma function, $d^{**da} * \text{incomplete gamma}(da,d) / \text{gamma}(da)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is GAMIT. | Class(es): C7e | Usage: D1 = DGAMIT (DA,D) | On-line doc: CALL GAMSDOC DGAMIT (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DGAMMA** Double precision gamma function. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is GAMMA. | Class(es): C7a | Usage: D1 = DGAMMA (D) | On-line doc: CALL GAMSDOC DGAMMA (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB | See also: DGAMLM
- DGAMR** Double precision reciprocal gamma function 1 / Gamma(d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is GAMR. | Class(es): C7a | Usage: D1 = DGAMR (D) | On-line doc: CALL GAMSDOC DGAMR (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DGAUSQ** Finds the abscissae and weights for Gauss Legendre quadrature on the interval (a b). | Proprietary double precision Fortran subprogram in PORT library. Single precision version is GAUSQ. | Class(es): H2c | Usage: CALL DGAUSQ (N,A,B,C,NU,X,W) | On-line doc: CALL GAMSDOC DGAUSQ (or @PRT PORT*DOC.DGAUSQ) | Access: LIB NBS*PORT
- DGBCO** Computes LU factorization of general double precision band matrix and estimates its condition. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SBCO. | Class(es):

- D2a2** | Usage: CALL DGBCO(ABD,LDA,N,ML,MU,IPVT,RCOND,Z) | On-line doc: CALL GAMSDOC DGBCO (or @PRT CMLIB*DOC.DGBCO/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB
- DGBDI** Uses LU factorization of general double precision band matrix to compute its determinant. (No provision for inverse computation.) | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SGBDI. | Class(es): D3a2 | Usage: CALL DGBDI(ABD,LDA,N,ML,MU,IPVT,DET) | On-line doc: CALL GAMSDOC DGBDI (or @PRT CMLIB*DOC.DGBDI/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DGBCO,DGBFA
- DGBFA** Computes LU factorization of general double precision band matrix. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SGBFA. | Class(es): D2a2 | Usage: CALL DGBFA(ABD,LDA,N,ML,MU,IPVT,INFO) | On-line doc: CALL GAMSDOC DGBFA (or @PRT CMLIB*DOC.DGBFA/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB
- DGBSL** Uses LU factorization of general double precision band matrix to solve systems. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SGBSL. | Class(es): D2a2 | Usage: CALL DGBSL(ABD,LDA,N,ML,MU,IPVT,B,JOB) | On-line doc: CALL GAMSDOC DGBSL (or @PRT CMLIB*DOC.DGBSL/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DGBCO,DGBFA
- DGEAR** Differential equation solver - variable order Adams predictor corrector method or Gears method. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): I1a2 I1a1b | Usage: CALL DGEAR (N,FCN,FCNJ,X,H,Y,XEND,TOL,METH,MITER,INDEX,IWK,WK, IER) | On-line doc: CALL GAMSDOC DGEAR (or @PRT IMSL*DOC.DGEAR) | Access: LIB NBS*IMSL
- DGECO** Compute LU factorization of general double precision matrix and estimate its condition. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SGECO. | Class(es): D2a1 | Usage: CALL DGECO(A,LDA,N,IPVT,RCOND,Z) | On-line doc: CALL GAMSDOC DGECO (or @PRT CMLIB*DOC.DGECO/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB
- DGEDI** Uses LU factorization of general double precision matrix to compute its determinant and/or inverse. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SGEDI. | Class(es): D3a1 D2a1 | Usage: CALL DGEDI(A,LDA,N,IPVT,DET,WORK,JOB) | On-line doc: CALL GAMSDOC DGEDI (or @PRT CMLIB*DOC.DGEDI/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DGECO,DGEFA
- DGEFA** Compute LU factorization of general double precision matrix. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SGEFA. | Class(es): D2a1 | Usage: CALL DGEFA(A,LDA,N,IPVT,INFO) | On-line doc: CALL GAMSDOC DGEFA (or @PRT CMLIB*DOC.DGEFA/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB
- DGEFS** Factors and solves a general NXN double precision system of linear equations. | Portable double precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. Single precision version is SGEFS. | Class(es): D2a1 | Usage: CALL DGEFS(A,LDA,N,V,ITASK,IND,WORK,IWORK) | On-line doc: CALL GAMSDOC DGEFS (or @PRT CMLIB*DOC.DGEFS/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F2/LINDRIVES | Access: LIB NBS*CMLIB
- DGESL** Uses LU factorization of general double precision matrix to solve systems. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SGESL. | Class(es): D2a1 | Usage: CALL DGESL(A,LDA,N,IPVT,B,JOB) | On-line doc: CALL GAMSDOC DGESL (or @PRT CMLIB*DOC.DGESL/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DGECO,DGEFA
- DGQ0IN** Finds the abscissae and weights for Gauss Laguerre quadrature on the interval (0 +infinity). | Proprietary double precision Fortran subprogram in PORT library. Single precision version is GQ0IN. | Class(es): H2c | Usage: CALL DGQ0IN (N,X,W) | On-line doc: CALL GAMSDOC DGQ0IN (or @PRT PORT*DOC.DGQ0IN) | Access: LIB NBS*PORT
- DGQM11** Finds the abscissae and weights for Gauss Legendre quadrature on the interval (-1,1). | Proprietary double precision Fortran subprogram in PORT library. Single precision version is GQM11. | Class(es): H2c | Usage: CALL DGQM11 (N,X,W) | On-line doc: CALL GAMSDOC DGQM11 (or @PRT PORT*DOC.DGQM11) | Access: LIB NBS*PORT
- DGTSL** Solve systems with tridiagonal double precision matrix. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SGTSL. | Class(es): D2a2a | Usage: CALL DGTSL(N,C,D,E,B,INFO) | On-line doc: CALL GAMSDOC DGTSL (or @PRT CMLIB*DOC.DGTSL/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB
- DHUMSL** Minimizes a general unconstrained objective function using (analytic) gradient and Hessian provided by the user. (Double precision version of HUMSL.) | Portable double precision Fortran subprogram in DNL2SN sublibrary of CMLIB library. Single precision version is HUMSL. | Class(es): G1b1c | Usage: CALL DHUMSL(N,D,X,CALCF,CALCGH,IV,LIV,LV,V,UIPARM,URPARI, UFPARM) | On-line doc: CALL GAMSDOC DHUMSL (or @PRT CMLIB*DOC.DHUMSL/DNL2SN) | Tests: CMLIB*TEST-SOURCE.\$F2/DNL2SN, CMLIB*TEST-SOURCE.\$Q2/DNL2SN | Access: LIB NBS*CMLIB
- DIFFERENCES** Computes differences between observations at a specified lag in a time series. | Command in MINITAB Proprietary interactive

system. Class(es): L10a | Usage: Differences [of lag K] for data in C, put into C | On-line doc: HELP DIFFERENCES (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

DIFFERENCES Computes all differences $X - Y$ for each value X in one vector and each value Y in a second vector (useful for nonparametric tests and confidence intervals). | Command in MINITAB Proprietary interactive system. Class(es): L2a | Usage: Differences between values in C and C, put into C [put indices into C and C] | On-line doc: HELP DIFFERENCES (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

DIFFERENTI Calculates 1st, 2nd, or 3rd derivative of given function F at X_0 to specified accuracy, program picks evaluation points. | Portable single precision Algol subprogram in OLIVER sublibrary of MATHWARE library. | Class(es): H1 | Usage: CALL DIFFERENTIATE(ORDER,X0,XMIN,XMAX,F,EPS,ACC,ETA,INF,SUP, DERIVATIVE, ERROR, FAILURE) | On-line doc: @PRT,S MATHWARE*OLIVER.DOC/ALGOL | Tests: MATHWARE*OLIVER.FTEST/ALGOL and MATHWARE*OLIVER.QTEST/ALGOL | Access: See individual sublibrary documentation

DINTRV Computes the index into a knot or breakpoint sequence corresponding to a given point X . | Portable double precision Fortran subprogram in DBSPLINE sublibrary of CMLIB library. Single precision version is INTRV. | Class(es): E3 K6 | Usage: CALL DINTRV(KT,LXT,X,ILD,ILEFT,MFLAG) | On-line doc: CALL GAMSDOC DINTRV (or @PRT CMLIB*DOC.DINTRV/DBSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/DBSPLINE | Access: LIB NBS*CMLIB

DISCR2 Discretizes the data in the vector X into NUMCLA classes. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L2a | Usage: CALL DISCR2(X,N,NUMCLA,Y) | On-line doc: CALL GAMSDOC DISCR2 (or @PRT DATAPAC*DOC.DISCR2) | Access: LIB NBS*DAPAC

DISCR3 Discretizes the data in the vector X into NUMCLA classes. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L2a | Usage: CALL DISCR3(X,N,NUMCLA,Y) | On-line doc: CALL GAMSDOC DISCR3 (or @PRT DATAPAC*DOC.DISCR3) | Access: LIB NBS*DAPAC

DISCRE Discretizes the data of the vector X according to class width. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L2a | Usage: CALL DISCRE(X,N,XMIN,XDEL,XMAX,Y) | On-line doc: CALL GAMSDOC DISCRE (or @PRT DATAPAC*DOC.DISCRE) | Access: LIB NBS*DAPAC

DL2SF Fits discrete data with a B-spline of order K , by least squares. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DDL2SF. | Class(es): K1a1a1 | Usage: CALL DL2SF (X,Y,NXY,K,T,NT,A) | On-line doc: CALL GAMSDOC DL2SF (or @PRT PORT*DOC.DL2SF) | Access: LIB NBS*PORT | See also: DSPLNE DSPLNI DSPLN1 DSPLN2 DSPLND

DL2SFF Obtains a weighted least square expansion of a known function in terms of B-splines of order K , at given mesh points. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is L2SFF. | Class(es): K1a1a1 | Usage: CALL DL2SFF (FW,K,T,NT,A) | On-line doc: CALL GAMSDOC DL2SFF (or @PRT PORT*DOC.DL2SFF) | Access: LIB NBS*PORT | See also: DSPLNE DSPLND DSPLNI DSPLN1 DSPLN2 DEESF DEESFI DEESFF DEESFI

DL2SFH Obtains a weighted least square expansion of a known function in and its derivatives in terms of B-splines of order K at given mesh points. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is L2SFH. | Class(es): K1a1a1 | Usage: CALL DL2SFH (FW,MD,K,T,NT,A) | On-line doc: CALL GAMSDOC DL2SFH (or @PRT PORT*DOC.DL2SFH) | Access: LIB NBS*PORT | See also: DSPLNE DSPLND DSPLNI DSPLN1 DSPLN2

DL2SW Fits discrete data with a B-spline of order k , by weighted least squares. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DDL2SW. | Class(es): K1a1a1 | Usage: CALL DL2SW (X,Y,NXY,W,K,T,NT,A) | On-line doc: CALL GAMSDOC DL2SW (or @PRT PORT*DOC.DL2SW) | Access: LIB NBS*PORT | See also: SPLNE SPLND SPLNI SPLN1 SPLN2

DLBETA Double precision Log Beta function, $\ln B(a,b)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is ALBETA. | Class(es): C7b | Usage: D = DLBETA (A,B) | On-line doc: CALL GAMSDOC DLBETA (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

DLGAMS Double precision Log abs gamma with sign of gamma, $G = \ln \text{abs Gamma}(d)$, $S = \text{sign Gamma}(d)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is ALGAMS. | Class(es): C7a | Usage: CALL DLGAMS (D,G,S) | On-line doc: CALL GAMSDOC DLGAMS (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

DLI Double precision logarithmic integral. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is ALI. | Class(es): C5 | Usage: D1 = DLI (D) | On-line doc: CALL GAMSDOC DLI (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

DLINEQ Solves a real system of linear equations, $AX=B$, where B is allowed to be a matrix or a vector. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is LINEQ. | Class(es): D2a1 | Usage: CALL DLINEQ (N, A, B, NB, X) | On-line doc: CALL GAMSDOC DLINEQ (or @PRT PORT*DOC.DLINEQ) | Access: LIB NBS*PORT

DLNGAM Double precision Log abs gamma, \ln of absolute value of Gamma(d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is ALNGAM. | Class(es): C7a | Usage: D1 = DLNGAM (D) | On-line doc: CALL GAMSDOC DLNGAM (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

DLNREL Double precision relative error logarithm, $\ln(1 + d)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is ALNREL. | Class(es): C4b | Usage: D1 = DLNREL (D) | On-line doc: CALL GAMSDOC DLNREL (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

DLSTSQ Finds the least squares solution of a system of linear equations, $AX=B$. B may be a matrix. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is LSTSQ. | Class(es): D0 | Usage: CALL DLSTSQ (MDIM,NDIM,M,N,A,B,NB,X)

- | On-line doc: CALL GAMSDOC DLSTSQ (or @PRT PORT*DOC.DLSTSQ) | Access: LIB NBS*PORT
- DLUMB** Given a basic mesh, this subdivides each interval uniformly for B-spline use. Multiplicities are allowed. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is LUMB. | Class(es): E3 K6 | Usage: CALL DLUMB (XB,NXB,N,K,X,NX) | On-line doc: CALL GAMSDOC DLUMB (or @PRT PORT*DOC.DLUMB) | Access: LIB NBS*PORT
- DLUMD** Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): E3 K6 | Usage: CALL DLUMD (XB,NXB,N,X,NX) | On-line doc: CALL GAMSDOC DLUMD (or @PRT PORT*DOC.DLUMD) | Access: LIB NBS*PORT
- DMLIN** Numerical integration of a function of several variables over a hyper-rectangle (Gaussian method). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): H2b1a1 | Usage: R = DMLIN(F,A,B,N,MAXFCN,AERR,RERR,IER) | On-line doc: CALL GAMSDOC DMLIN (or @PRT IMSL*DOC.DMLIN) | Access: LIB NBS*IMSL
- DMNPB** Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each interval. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is MNPB. | Class(es): E3 K6 | Usage: CALL DMNPB (X,NX,N,K,T,NT) | On-line doc: CALL GAMSDOC DMNPB (or @PRT PORT*DOC.DMNPB) | Access: LIB NBS*PORT
- DNBCO** Factors a double precision band matrix by Gaussian elimination and estimates the condition of the matrix. | Portable double precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. Single precision version is SNBCO. | Class(es): D2a2 | Usage: CALL DNBCO(ABE,LDA,N,ML,MU,IPVT,RCOND,Z) | On-line doc: CALL GAMSDOC DNBCO (or @PRT CMLIB*DOC.DNBCO/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F2/LINDRIVES | Access: LIB NBS*CMLIB
- DNBDI** Computes determinant of double precision band matrix using factors previously computed. | Portable double precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. Single precision version is SNBDI. | Class(es): D3a2 | Usage: CALL DNBDI(ABE,LDA,N,ML,MU,IPVT,DET) | On-line doc: CALL GAMSDOC DNBDI (or @PRT CMLIB*DOC.DNBDI/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F2/LINDRIVES | Access: LIB NBS*CMLIB | See also: DNBCO, DNBFA
- DNBFA** Factors a double precision band matrix by Gaussian elimination. | Portable double precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. Single precision version is SNBFA. | Class(es): D2a2 | Usage: CALL DNBFA(ABE,LDA,N,ML,MU,IPVT,INFO) | On-line doc: CALL GAMSDOC DNBFA (or @PRT CMLIB*DOC.DNBFA/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F2/LINDRIVES | Access: LIB NBS*CMLIB
- DNBFS** Factors and solves general nonsymmetric banded double precision system of linear equations. | Portable double precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. Single precision version is SNBFS. | Class(es): D2a2 | Usage: CALL DNBFS(ABE,LDA,N,ML,MU,V,ITASK,IND,WORK,IWORK) | On-line doc: CALL GAMSDOC DNBFS (or @PRT CMLIB*DOC.DNBFS/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F2/LINDRIVES | Access: LIB NBS*CMLIB
- DNBSL** Solves a double precision band system of equations using previously computed factors. | Portable double precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. Single precision version is SNBSL. | Class(es): D2a2 | Usage: CALL DNBSL(ABE,LDA,N,ML,MU,IPVT,B,JOB) | On-line doc: CALL GAMSDOC DNBSL (or @PRT CMLIB*DOC.DNBSL/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F2/LINDRIVES | Access: LIB NBS*CMLIB | See also: DNBCO, DNBFA
- DNL2S1** Minimizes a nonlinear sum of squares using both residual and gradient values supplied by the user. (Double precision version of NL2S1.). | Portable double precision Fortran subprogram in DNL2SN sublibrary of CMLIB library. Single precision version is NL2S1. | Class(es): L8g1b L8g2b K1b1a2 | Usage: CALL DNL2S1(N,P,X,CALCR,CALCJ,IV,LIV,LV,V,UIPARM,URPARM,UFPARM) | On-line doc: CALL GAMSDOC DNL2S1 (or @PRT CMLIB*DOC.DNL2S1/DNL2SN) | Tests: CMLIB*TEST-SOURCE.\$F1/DNL2SN, CMLIB*TEST-SOURCE.\$Q1/DNL2SN | Access: LIB NBS*CMLIB
- DNL2SN** Minimizes a nonlinear sum of squares using residual values only. (Double precision version of NL2SN.). | Portable double precision Fortran subprogram in DNL2SN sublibrary of CMLIB library. Single precision version is NL2SN. | Class(es): L8g1a L8g2a K1b1a1 | Usage: CALL DNL2SN(N,P,X,CALCR,IV,LIV,LV,V,UIPARM,URPARM,UFPARM) | On-line doc: CALL GAMSDOC DNL2SN (or @PRT CMLIB*DOC.DNL2SN/DNL2SN) | Tests: CMLIB*TEST-SOURCE.\$F1/DNL2SN, CMLIB*TEST-SOURCE.\$Q1/DNL2SN | Access: LIB NBS*CMLIB
- DNRM2** Compute the Euclidean length or L2 norm of a double precision vector. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. Single precision version is SNRM2. | Class(es): D1a3b | Usage: D = DNRM2(N,DX,INCX) | On-line doc: CALL GAMSDOC DNRM2 (or @PRT CMLIB*DOC.DNRM2/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- DNRM2** Finds the length (Euclidean norm) of a vector, without underflow or overflow. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is SNRM2. | Class(es): D1a3b | Usage: D = DNRM2 (N, X, INCX) | On-line doc: CALL GAMSDOC DNRM2 (or @PRT PORT*DOC.DNRM2) | Access: LIB NBS*PORT
- DODEQ** Finds the integral of a set of functions over the same interval by using the differential equation solver ODES1. For smooth functions. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ODEQ. | Class(es): H2a1a1 | Usage: CALL DODEQ (N, F, A, B, EPS, ANS) | On-line doc: CALL GAMSDOC DODEQ (or @PRT PORT*DOC.DODEQ) | Access: LIB NBS*PORT
- DODES** Solves an initial value problem for a system of ordinary differential equations. Easy to use. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ODES. | Class(es): I1a1c | Usage: CALL DODES (F,X,NX,TSTART,TSTOP,DT,ERRPAR,HANDLE) | On-line doc: CALL GAMSDOC DODES (or @PRT PORT*DOC.DODES) | Access: LIB NBS*PORT
- DODES1** Solves an initial value problem for a system of ordinary differential equations. Allows great flexibility and user control. | Proprietary

- double precision Fortran subprogram in PORT library. Single precision version is ODES1. | Class(es): 11a1c | Usage: CALL DODES1 (F,X,NX,TSTART,TSTOP,DT,ERROR,ERRPAR,HANDLE,GLBMAX, ERPUTS) | On-line doc: CALL GAMSDOC DODES1 (or @PRT PORT*DOC.DODES1) | Access: LIB NBS*PORT | See also: DODESH DODESE
- DODESE** Standard error subprogram for the routine ODES1. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ODESE. | Class(es): 11c | Usage: L = DODESE (X,NX,T,DT,ERRPAR,ERPUTS,E) | On-line doc: CALL GAMSDOC DODESE (or @PRT PORT*DOC.DODESE) | Access: LIB NBS*PORT | See also: DODES1
- DODESH** Default HANDLE routine for ODES. Used to access the results at the end of each integration time step. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ODESH. | Class(es): 11c | Usage: CALL DODESH (T0,X0,T1,X1,NX,DT,TSTOP,E) | On-line doc: CALL GAMSDOC DODESH (or @PRT PORT*DOC.DODESH) | Access: LIB NBS*PORT | See also: DODES
- DORTH** Evaluates a polynomial expressed as a sum of general orthogonal polynomials. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ORTHP. | Class(es): C3 | Usage: D = DORTH (N, ALPHA, X, A, B, C) | On-line doc: CALL GAMSDOC DORTH (or @PRT PORT*DOC.DORTH) | Access: LIB NBS*PORT
- DPBCO** Compute LU factorization of double precision positive definite band matrix and estimate its condition. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SPBCO. | Class(es): D2b2 | Usage: CALL DPBCO(ABD,LDA,N,M,RCOND,Z,INFO) | On-line doc: CALL GAMSDOC DPBCO (or @PRT CMLIB*DOC.DPBCO/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB
- DPBDI** Use LU factorization of double precision positive definite band matrix to compute determinant. (No provision for inverse.). | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SPBDI. | Class(es): D3b2 | Usage: CALL DPBDI(ABD,LDA,N,M,DET) | On-line doc: CALL GAMSDOC DPBDI (or @PRT CMLIB*DOC.DPBDI/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DPBCO DPBFA
- DPBFA** Computes LU factorization of double precision positive definite band matrix. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SPBFA. | Class(es): D2b2 | Usage: CALL DPBFA(ABD,LDA,N,M,INFO) | On-line doc: CALL GAMSDOC DPBFA (or @PRT CMLIB*DOC.DPBFA/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB
- DPBSL** Uses LU factorization of double precision positive definite band matrix to solve systems. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SPBSL. | Class(es): D2b2 | Usage: CALL DPBSL(ABD,LDA,N,M,B) | On-line doc: CALL GAMSDOC DPBSL (or @PRT CMLIB*DOC.DPBSL/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DPBCO DPBFA
- DPDES** Solve a system of partial differential equations of the form $UT = FCN(X,T,U,UX, UXX)$ using the method of lines with cubic Hermite polynomials. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): I2a1a I2a2 | Usage: CALL DPDES (NPDES,FCN,BNDRY,T,H,TEND,X,Y,IY,NX,TOL,INDEX,WK, IER) | On-line doc: CALL GAMSDOC DPDES (or @PRT IMSL*DOC.DPDES) | Access: LIB NBS*IMSL
- DPFQAD** Integrates function times derivative of B-spline from X1 to X2. The B-spline is in piecewise polynomial representation. | Portable double precision Fortran subprogram in DBSPLINE sublibrary of CMLIB library. Single precision version is PFQAD. | Class(es): H2a2a1 E3 K6 | Usage: CALL DPFQAD(F,LDC,X1,LX1,K,ID,X1,X2,TOL,QUAD,IERR) | On-line doc: CALL GAMSDOC DPFQAD (or @PRT CMLIB*DOC.DPFQAD/DBSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/DBSPLINE | Access: LIB NBS*CMLIB
- DPOCH** Double precision Pochhammer's generalized symbol, $= \Gamma(a+d) / \Gamma(a)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is POCH. | Class(es): C1 C7a | Usage: D1 = DPOCH (A,D) | On-line doc: CALL GAMSDOC DPOCH (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DPOCH1** Double precision Pochhammer's symbol from first order, $= ((a) \text{ sub } d - 1) / d$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is POCH1. | Class(es): C1 C7a | Usage: D1 = DPOCH1 (A,D) | On-line doc: CALL GAMSDOC DPOCH1 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DPOCO** Use Cholesky algorithm to factor double precision positive definite matrix and estimate its condition. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SPOCO. | Class(es): D2b1b | Usage: CALL DPOCO(A,LDA,N,RCOND,Z,INFO) | On-line doc: CALL GAMSDOC DPOCO (or @PRT CMLIB*DOC.SUMMARY/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB
- DPODI** Use factorization of double precision positive definite matrix to compute determinant and/or inverse. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SPODI. | Class(es): D2b1b D3b1b | Usage: CALL DPODI(A,LDA,N,DET,JOB) | On-line doc: CALL GAMSDOC DPODI (or @PRT CMLIB*DOC.DPODI/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DPOCO DPOFA
- DPOFA** Use Cholesky algorithm to factor double precision positive definite matrix. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SPOFA. | Class(es): D2b1b | Usage: CALL DPOFA(A,LDA,N,INFO) | On-line doc: CALL GAMSDOC DPOFA (or @PRT CMLIB*DOC.DPOFA/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB

- DPOFS** Factors and solves a positive definite symmetric double precision system of equations. | Portable double precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. Single precision version is SPOFS. | Class(es): D2b1b | Usage: CALL DPOFS(A,LDA,N,V,ITASK,IND,WORK) | On-line doc: CALL GAMSDOC DPOFS (or @PRT CMLIB*DOC.DPOFS/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F2/LINDRIVES | Access: LIB NBS*CMLIB
- DPOSL** Use factorisation of double precision positive definite matrix to solve systems. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SPOSL. | Class(es): D2b1b | Usage: CALL DPOSL(A,LDA,N,B) | On-line doc: CALL GAMSDOC DPOSL (or @PRT CMLIB*DOC.DPOSL/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DPOCO DPOFA
- DPPCO** Use Cholesky algorithm to factor double precision positive definite matrix stored in packed form and estimate its condition. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SPPCO. | Class(es): D2b1b | Usage: CALL DPPCO(AP,N,RCOND,Z,INFO) | On-line doc: CALL GAMSDOC DPPCO (or @PRT CMLIB*DOC.DPPCO/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB
- DPPDI** Use factorisation of double precision positive definite matrix stored in packed form to compute determinant and/or inverse. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SPPDI. | Class(es): D2b1b D3b1b | Usage: CALL DPPDI(AP,N,DET,JOB) | On-line doc: CALL GAMSDOC DPPDI (or @PRT CMLIB*DOC.DPPDI/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DPPCO DPPFA
- DPPFA** Use Cholesky algorithm to factor double precision positive definite matrix stored in packed form. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SPPFA. | Class(es): D2b1b | Usage: CALL DPPFA(AP,N,INFO) | On-line doc: CALL GAMSDOC DPPFA (or @PRT CMLIB*DOC.DPPFA/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB
- DPPQAD** Computes the integral of a B-spline from X1 to X2. The B-spline must be in piecewise polynomial form. | Portable double precision Fortran subprogram in DBSPLINE sublibrary of CMLIB library. Single precision version is PPQAD. | Class(es): H2a1a1 E8 K6 | Usage: CALL DPPQAD(LDC,C,XI,LXI,K,X1,X2,PQUAD) | On-line doc: CALL GAMSDOC DPPQAD (or @PRT CMLIB*DOC.DPPQAD/DBSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/DBSPLINE | Access: LIB NBS*CMLIB
- DPPSL** Use factorisation of double precision positive definite matrix stored in packed form to solve systems. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SPPSL. | Class(es): D2b1b | Usage: CALL DPPSL(AP,N,B) | On-line doc: CALL GAMSDOC DPPSL (or @PRT CMLIB*DOC.DPPSL/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DPPCO DPPFA
- DPPVAL** Calculates (at X) the value of the IDERIV-th derivative of the B-spline from its piecewise polynomial representation. | Portable double precision Fortran subprogram in DBSPLINE sublibrary of CMLIB library. Single precision version is PPVAL. | Class(es): E8 K6 | Usage: CALL DPPVAL(LDC,C,XI,LXI,K,IDERIV,X,INPPV) | On-line doc: CALL GAMSDOC DPPVAL (or @PRT CMLIB*DOC.DPPVAL/DBSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/DBSPLINE | Access: LIB NBS*CMLIB
- DPSI** Double precision Psi (digamma), $= \Gamma'(d) / \Gamma(d)$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is PSI. | Class(es): C7c | Usage: D1 = DPSI(D) | On-line doc: CALL GAMSDOC DPSI (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DPTSL** Decomposes double precision symmetric positive definite tridiagonal matrix and simultaneously solve a system. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SPTSL. | Class(es): D2b2a | Usage: CALL DPTSL(N,D,E,B) | On-line doc: CALL GAMSDOC DPTSL (or @PRT CMLIB*DOC.DPTSL/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB
- DPUMB** Given a basic mesh, this subdivides each interval into a uniform but variable number of points. Multiplicities can occur. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is PUMB. | Class(es): E8 K6 | Usage: CALL DPUMB (XB,NXB,NA,K,X,NX) | On-line doc: CALL GAMSDOC DPUMB (or @PRT PORT*DOC.DPUMB) | Access: LIB NBS*PORT
- DPUMD** Given a basic mesh, this subdivides each interval into a uniform but variable number of points. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is PUMD. | Class(es): E8 K6 | Usage: CALL DPUMD (XB,NXB,NA,X,NX) | On-line doc: CALL GAMSDOC DPUMD (or @PRT PORT*DOC.DPUMD) | Access: LIB NBS*PORT
- DQAG** Automatic adaptive integrator, will handle many non-smooth integrands using Gauss Kronrod formulas. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QAG. | Class(es): H2a1a1 | Usage: CALL DQAG(F,A,B,EPSABS,EPSREL,KEY,RESULT,ABSERR,NEVAL,IER, LIMIT,LENW, LAST, IWORK WORK) | On-line doc: CALL GAMSDOC DQAG (or @PRT CMLIB*DOC.DQAG/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.DQAG/QUADDP, CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB
- DQAGE** Automatic adaptive integrator, can handle most non-smooth functions also provides more information than DQAG. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QAGE. | Class(es): H2a1a1 | Usage: CALL DQAGE(F,A,B,EPSABS,EPSREL,KEY,LIMIT,RESULT,ABSERR,NEVAL, IER,ALIST, BLIST, RLIST,ELIST,IORD,LAST) | On-line doc: CALL GAMSDOC DQAGE (or @PRT CMLIB*DOC.DQAGE/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB

- DQAGI** Automatic adaptive integrator for semi-infinite or infinite intervals. Uses nonlinear transformation and extrapolation. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QAGI. | Class(es): H2a3a1 H2a4a1 | Usage: CALL DQAGI(F,BOUND,INF,EPSABS,EPSREL,RESULT,ABSERR,NEVAL,IER, LIMIT, LENW, LAST, IWORK, WORK) | On-line doc: CALL GAMSDOC DQAGI (or @PRT CMLIB*DOC.DQAGI/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.DQAGI/QUADDP, CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB
- DQAGIE** Automatic integrator for semi-infinite or infinite intervals and general integrands, provides more information than DQAGI. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QAGIE. | Class(es): H2a3a1 H2a4a1 | Usage: CALL DQAGIE(F,BOUND,INF,EPSABS,EPSREL,LIMIT,RESULT,ABSERR, NEVAL,IER,ALIST, BLIST,RLIST,ELIST,IORD, LAST) | On-line doc: CALL GAMSDOC DQAGIE (or @PRT CMLIB*DOC.DQAGIE/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB
- DQAGP** Automatic adaptive integrator, allows user to specify location of singularities or difficulties of integrand, uses extrapolation. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QAGP. | Class(es): H2a2a1 | Usage: CALL DQAGP(F,A,B,NPTS2,POINTS,EPSABS,EPSREL,RESULT,ABSERR, NEVAL,IER, LENIW,LENW, IWORK,WORK) | On-line doc: CALL GAMSDOC DQAGP (or @PRT CMLIB*DOC.DQAGP/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.DQAGP/QUADDP, CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB
- DQAGPE** Automatic adaptive integrator for function with user specified endpoint singularities, provides more information than DQAGP. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QAGPE. | Class(es): H2a2a1 | Usage: CALL DQAGPE(F,A,B,NPTS2,POINTS,EPSABS,EPSREL,LIMIT,RESULT, ABSERR,NEVAL,IER, ALIST,BLIST,RLIST,ELIST,PTS,IORD,LEVEL,NDIN, LAST) | On-line doc: CALL GAMSDOC DQAGPE (or @PRT CMLIB*DOC.DQAGPE/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB
- DQAGS** Automatic adaptive integrator, will handle most non-smooth integrands including those with endpoint singularities, uses extrapolation. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QAGS. | Class(es): H2a1a1 | Usage: CALL DQAGS(F,A,B,EPSABS,EPSREL,RESULT,ABSERR,NEVAL,IER,LIMIT, LENW, LAST,IWORK, WORK) | On-line doc: CALL GAMSDOC DQAGS (or @PRT CMLIB*DOC.DQAGS/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.DQAGS/QUADDP, CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB
- DQAGSE** Automatic adaptive integrator, can handle integrands with endpoint singularities provides more information than DQAGS. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QAGSE. | Class(es): H2a1a1 | Usage: CALL DQAGSE(F,A,B,EPSABS,EPSREL,LIMIT,RESULT,ABSERR,NEVAL,IER, ALIST, BLIST,RLIST,ELIST,IORD, LAST) | On-line doc: CALL GAMSDOC DQAGSE (or @PRT CMLIB*DOC.DQAGSE/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB
- DQAWC** Cauchy principal value integrator, using adaptive Clenshaw Curtis method (real Hilbert transform). | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QAWC. | Class(es): H2a2a1 J4 | Usage: CALL DQAWC(F,A,B,C,EPSABS,EPSREL,RESULT,ABSERR,NEVAL,IER,LIMIT, LENW, LAST,IWORK, WORK) | On-line doc: CALL GAMSDOC DQAWC (or @PRT CMLIB*DOC.DQAWC/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.DQAWC/QUADDP, CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB
- DQAWCE** Cauchy Principal value integrator, provides more information than DQAWC (real Hilbert transform). | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QAWCE. | Class(es): H2a2a1 J4 | Usage: CALL DQAWCE(F,A,B,C,EPSABS,EPSREL,LIMIT,RESULT,ABSERR,NEVAL, IER,ALIST, BLIST,RLIST,ELIST,IORD, LAST) | On-line doc: CALL GAMSDOC DQAWCE (or @PRT CMLIB*DOC.DQAWCE/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB
- DQAWF** Automatic integrator for Fourier integrals on (a, infinity) with factors SIN(OMEGA*X), COS(OMEGA*X) by integrating between zeros. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QAWF. | Class(es): H2a3a1 | Usage: CALL DQAWF(F,A,OMEGA,INTEGR,EPSABS,RESULT,ABSERR,NEVAL,IER, LIMLST, LST,LENIW, MAXP1,LENW,IWORK,WORK) | On-line doc: CALL GAMSDOC DQAWF (or @PRT CMLIB*DOC.DQAWF/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.DQAWF/QUADDP, CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB
- DQAWFE** Automatic integrator for Fourier integrals, with SIN(OMEGA*X) factor on (A, INFINITY), provides more information than DQAWF. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QAWFE. | Class(es): H2a3a1 | Usage: CALL DQAWFE(F,A,OMEGA,INTEGR,EPSABS,LIMLST,LIMIT,MAXP1,RESULT, ABSERR,NEVAL,IER, RSLST,ERLST,IERLST,LST,ALIST,BLIST,RLIST,ELIST,IORD, NNLOG,CHEBMO) | On-line doc: CALL GAMSDOC DQAWFE (or @PRT CMLIB*DOC.DQAWFE/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB
- DQAWO** Automatic adaptive integrator for integrands with oscillatory sin or cosine factor. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QAWO. | Class(es): H2a2a1 | Usage: CALL DQAWO(F,A,B,OMEGA,INTEGR,EPSABS,EPSREL,RESULT,ABSERR, NEVAL,IER, LENIW,MAXP1 LENW, LAST,IWORK,WORK) | On-line doc: CALL GAMSDOC DQAWO (or @PRT CMLIB*DOC.DQAWO/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.DQAWO/QUADDP, CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB
- DQAWOE** Automatic integrator for integrands with explicit oscillatory sin or cosine factor, provides more information than DQAWO. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QAWOE. | Class(es): H2a2a1 | Usage: CALL DQAWOE(F,A,B,OMEGA,INTEGR,EPSABS,EPSREL,LIMIT,ICALL,MAXP1, RESULT,ABSERR,

NEVAL,IER,ALIST,BLIST,RLIST,ELIST,IORD,NNLOG,MOMCOM,CHEBMO) | On-line doc: CALL GAMSDOC DQAWOE (or @PRT CMLIB*DOC.DQAWOE/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB

DQAWS Automatic integrator for functions with explicit algebraic and/or logarithmic endpoint singularities. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QAWS. | Class(es): H2a2a1 | Usage: CALL DQAWS(F,A,B,ALFA,BETA,INTEGR,EPSABS,EPSREL,RESULT,ABSERR,NEVAL,IER,LIMIT,LENW,LAST,IWORK,WORK) | On-line doc: CALL GAMSDOC DQAWS (or @PRT CMLIB*DOC.DQAWS/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.DQAWS/QUADDP, CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB

DQAWSE Automatic integrator for integrands with explicit algebraic and/or logarithmic endpoint singularities, more information than DQAWS. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QAWSE. | Class(es): H2a2a1 | Usage: CALL DQAWSE(F,A,B,ALFA,BETA,INTEGR,EPSABS,EPSREL,LIMIT,RESULT,ABSERR,NEVAL,IER,ALIST,BLIST,RLIST,ELIST,IORD,LAST) | On-line doc: CALL GAMSDOC DQAWSE (or @PRT CMLIB*DOC.DQAWSE/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB

DQC25C Uses 25 point Clenshaw-Curtis formula to estimate integral of F^*W where $W=1/(X-C)$. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QC25C. | Class(es): H2a2a2 J4 | Usage: CALL DQC25C(F,A,B,C,RESULT,ABSERR,KRUL,NEVAL) | On-line doc: CALL GAMSDOC DQC25C (or @PRT CMLIB*DOC.DQC25C/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB

DQC25F Clenshaw-Curtis integration rule for function with cos or sin factor, also uses Gauss Kronrod formula. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QC25F. | Class(es): H2a2a2 | Usage: CALL DQC25F(F,A,B,OMEGA,INTEGR,NRMOM,MAXP1,KSAVE,RESULT,ABSERR,NEVAL,RESABS,RESASC,MOMCOM,CHEBMO) | On-line doc: CALL GAMSDOC DQC25F (or @PRT CMLIB*DOC.DQC25F/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB

DQC25S Estimates integral of function with algebraico-logarithmic singularities with 25 point Clenshaw-Curtis formula and gives error est. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QC25S. | Class(es): H2a2a2 | Usage: CALL DQC25S(F,A,B,BL,BR,ALFA,BETA,RI,RJ,RG,RH,RESULT,ABSERR,RESASC) INTEGR,NEV) | On-line doc: CALL GAMSDOC DQC25S (or @PRT CMLIB*DOC.DQC25S/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB

DQDOTA Compute dot product of 2 double precision vectors plus a double precision constant plus extended precision constant. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. | Class(es): D1a4 | Usage: D = DQDOTA(N,DB,QC,DX,INCX,DY,INCY) | On-line doc: CALL GAMSDOC DQDOTA (or @PRT CMLIB*DOC.DQDOTA/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

DQDOTI Compute in extended precision dot product of 2 double precision vectors plus d.p. constant. Result is double precision. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. | Class(es): D1a4 | Usage: D = DQDOTI(N,DB,QC,DX,INCX,DY,INCY) | On-line doc: CALL GAMSDOC DQDOTI (or @PRT CMLIB*DOC.DQDOTI/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

DQK15 Evaluates integral of given function on an interval with a 15 point Gauss Kronrod formula and returns error estimate. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QK15. | Class(es): H2a1a2 | Usage: CALL DQK15(F,A,B,RESULT,ABSERR,RESABS,RESASC) | On-line doc: CALL GAMSDOC DQK15 (or @PRT CMLIB*DOC.DQK15/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB

DQK15I Evaluates integral of given function on semi-infinite or infinite interval with a transformed 15 point Gauss Kronrod formula and gives error est. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QK15I. | Class(es): H2a3a2 H2a4a2 | Usage: CALL DQK15I(F,BOUN,INF,A,B,RESULT,ABSERR,RESABS,RESASC) | On-line doc: CALL GAMSDOC DQK15I (or @PRT CMLIB*DOC.DQK15I/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB

DQK15W Evaluates integral of given function times arbitrary weight function on interval with 15 point Gauss Kronrod formula and gives error estimate. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QK15W. | Class(es): H2a2a2 | Usage: CALL DQK15W(F,W,P1,P2,P3,P4,KP,A,B,RESULT,ABSERR,RESABS,RESASC) | On-line doc: CALL GAMSDOC DQK15W (or @PRT CMLIB*DOC.DQK15W/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB

DQK21 Evaluates integral of given function on an interval with a 21 point Gauss Kronrod formula and returns error estimate. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QK21. | Class(es): H2a1a2 | Usage: CALL DQK21(F,A,B,RESULT,ABSERR,RESABS,RESASC) | On-line doc: CALL GAMSDOC DQK21 (or @PRT CMLIB*DOC.DQK21/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB

DQK31 Evaluates integral of given function on an interval with a 31 point Gauss Kronrod formula and returns error estimate. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QK31. | Class(es): H2a1a2 | Usage: CALL DQK31(F,A,B,RESULT,ABSERR,RESABS,RESASC) | On-line doc: CALL GAMSDOC DQK31 (or @PRT CMLIB*DOC.DQK31/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB

DQK41 Evaluates integral of given function on an interval with a 41 point Gauss Kronrod formula and returns error estimate. |

Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QK41. | Class(es): H2a1a2 | Usage: CALL DQK41(F,A,B,RESULT,ABSERR,RESABS,RESASC) | On-line doc: CALL GAMSDOC DQK41 (or @PRT CMLIB*DOC.DQK41/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB

DQK51 Evaluates integral of given function on an interval with a 51 point Gauss Kronrod formula and returns error estimate. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QK51. | Class(es): H2a1a2 | Usage: CALL DQK51(F,A,B,RESULT,ABSERR,RESABS,RESASC) | On-line doc: CALL GAMSDOC DQK51 (or @PRT CMLIB*DOC.DQK51/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB

DQK61 Evaluates integral of given function on an interval with a 61 point Gauss Kronrod formula and returns error estimate. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QK61. | Class(es): H2a1a2 | Usage: CALL DQK61(F,A,B,RESULT,ABSERR,RESABS,RESASC) | On-line doc: CALL GAMSDOC DQK61 (or @PRT CMLIB*DOC.DQK61/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB

DQMOMO Computes integral of k-th degree Tchebycheff polynomial times selection of functions with various singularities. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QMOMO. | Class(es): H2a2a1 C3a2 | Usage: CALL DQMOMO(ALFA,BETA,RI,RJ,RG,RH,INTEGR) | On-line doc: CALL GAMSDOC DQMOMO (or @PRT CMLIB*DOC.DQMOMO/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB

DQNG Automatic non-adaptive integrator for smooth functions, using Gauss Kronrod Patterson formulas. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QNG. | Class(es): H2a1a1 | Usage: CALL DQNG(F,A,B,EPSABS,EPSREL,RESULT,ABSERR,NEVAL,IER) | On-line doc: CALL GAMSDOC DQNG (or @PRT CMLIB*DOC.DQNG/QUADDP and CMLIB*DOC.SUMMARY/QUADDP) | Tests: CMLIB*TEST-SOURCE.DQNG/QUADDP, CMLIB*TEST-SOURCE.\$Q/QUADDP | Access: LIB NBS*CMLIB

DQRANK For solving linear systems in least squares sense. Computes the QR decomposition of matrix using LINPACK subroutines. | Portable double precision Fortran subprogram in DQRLSS sublibrary of CMLIB library. Single precision version is SQRANK. | Class(es): D5 | Usage: CALL DQRANK(A,LDA,M,N,TOL,KR,JPVT,QRAUX,WORK) | On-line doc: CALL GAMSDOC DQRANK (or @PRT CMLIB*DOC.SUMMARY/DQRLSS and CMLIB*DOC.DQRANK/DQRLSS) | Access: LIB NBS*CMLIB

DQRDC Compute QR decomposition of general double precision matrix. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SQRDC. | Class(es): D5 | Usage: CALL DQRDC(X,LDX,N,P,QRAUX,JPVT,WORK,JOB) | On-line doc: CALL GAMSDOC DQRDC (or @PRT CMLIB*DOC.DQRDC/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB

DQRLSS Solves linear least squares problems of the form $Ax=b$. Calls LINPACK subroutines. Easy to use. | Portable double precision Fortran subprogram in DQRLSS sublibrary of CMLIB library. Single precision version is SQRSS. | Class(es): D9 | Usage: CALL DQRLSS(A,LDA,M,N,KR,B,X,RSD,JPVT,QRAUX) | On-line doc: CALL GAMSDOC DQRLSS (or @PRT CMLIB*DOC.DQRLSS/DQRLSS) | Tests: CMLIB*TEST-SOURCE.\$F/DQRLSS, CMLIB*TEST-SOURCE.\$Q/DQRLSS | Access: LIB NBS*CMLIB | See also: DQRANK FACTORS MATRIX, SQRSS SOLVES.

DQRSL Applies the output of DQRDC to compute coordinate transformations, projections, and least squares solutions (general double precision matrix). | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SQRSL. | Class(es): D9 D2a1 | Usage: CALL DQRSL(X,LDX,N,K,QRAUX,Y,QY,QTY,B,RSD,XB,JOB,INFO) | On-line doc: CALL GAMSDOC DQRSL (or @PRT CMLIB*DOC.DQRSL/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DQRDC

DQUAD Finds the integral of a general user defined EXTERNAL function by an adaptive technique to given absolute accuracy. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is QUAD. | Class(es): H2a1a1 | Usage: CALL DQUAD (F,A,B,EPS,ANS,ERREST) | On-line doc: CALL GAMSDOC DQUAD (or @PRT PORT*DOC.DQUAD) | Access: LIB NBS*PORT

DRANDOM Generates K pseudo-random numbers from a user-specified discrete distribution. | Command in MINITAB Proprietary interactive system. | Class(es): L6a4 | Usage: DRANDOM K observations, values in C, probabilities in C, put into C | On-line doc: HELP DRANDOM (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

DREBS Differential equation solver extrapolation method. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): 11a1c | Usage: CALL DREBS (FCN,Y,X,N,JM,IND,JSTART,H,HMIN,TOL,R,S,WK,IER) | On-line doc: CALL GAMSDOC DREBS (or @PRT IMSL*DOC.DREBS) | Access: LIB NBS*IMSL

DRLTR An auxiliary routine for use together with FFT to transform 2N real data points. Uses less storage than FFTR. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is RLTR. | Class(es): J1a1 | Usage: CALL DRLTR (A,B,N,ISN) | On-line doc: CALL GAMSDOC DRLTR (or @PRT PORT*DOC.DRLTR) | Access: LIB NBS*PORT | See also: DFFT

DROT Apply Givens plane rotation to double precision vector. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. Single precision version is SROT. | Class(es): D1a8 | Usage: CALL DROT(N,DX,INCX,DY,INCY,DC,DS) | On-line doc: CALL GAMSDOC DROT (or @PRT CMLIB*DOC.DROT/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

DROTG Construct Givens plane rotation of double precision matrix. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. Single precision version is SROTG. | Class(es): D1b10 | Usage: CALL DROTG(DA,DB,DC,DS) | On-line doc: CALL

- GAMSDOC DROTG (or @PRT CMLIB*DOC.DROTG/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- DROTM** Apply modified Givens plane rotation to double precision vector. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. Single precision version is SROTM. | Class(es): D1a8 | Usage: CALL DROTM(N,DX,INCX,DY,INCY,DPARAM) | On-line doc: CALL GAMSDOC DROTM (or @PRT CMLIB*DOC.DROTM/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- DROTMG** Construct modified Givens plane rotation of double precision matrix. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. Single precision version is SROTMG. | Class(es): D1b10 | Usage: CALL DROTMG(DD1,DD2,DB1,DB2,DPARAM) | On-line doc: CALL GAMSDOC DROTMG (or @PRT CMLIB*DOC.DROTMG/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- DRPOLY** Finds zeros of a polynomial with real coefficients. Output zeros are in a pair of arrays, for real and imaginary part. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is RPLOY. | Class(es): F1a1a | Usage: CALL DRPOLY (DEGREE,COEFF,ZEROR,ZEROI) | On-line doc: CALL GAMSDOC DRPOLY (or @PRT PORT*DOC.DRPOLY) | Access: LIB NBS*PORT
- DRQUAD** Finds the integral of a general user defined EXTERNAL function by an adaptive technique. Combined absolute and relative error control. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is RQUAD. | Class(es): H2a1a1 | Usage: CALL DRQUAD (F,A,B,EPSABS,EPSREL,ANS,ERREST) | On-line doc: CALL GAMSDOC DRQUAD (or @PRT PORT*DOC.DRQUAD) | Access: LIB NBS*PORT
- DRVTE** Calculate first, second, or third derivative of a user-supplied function. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): H1 | Usage: D = DRVTE(F,N,X,H,ERR,IER) | On-line doc: CALL GAMSDOC DRVTE (or @PRT IMSL*DOC.DRVTE) | Access: LIB NBS*IMSL
- DSCAL** Compute a constant times a vector, both double precision. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. Single precision version is SSCAL. | Class(es): D1a6 | Usage: CALL DSCAL(N,DA,DX,INCX) | On-line doc: CALL GAMSDOC DSCAL (or @PRT CMLIB*DOC.DSCAL/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- DSDOT** Compute single precision dot product using double precision accumulation. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. | Class(es): D1a4 | Usage: D = DSDOT(N,SX,INCX,SY,INCY) | On-line doc: CALL GAMSDOC DSDOT (or @PRT CMLIB*DOC.DSDOT/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- DSICO** Computes factorization of double precision symmetric indefinite matrix and estimate its condition. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SSICO. | Class(es): D2b1a | Usage: CALL DSICO(A,LDA,N,KPVT,RCOND,Z) | On-line doc: CALL GAMSDOC DSICO (or @PRT CMLIB*DOC.DSICO/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB
- DSIDI** Use factorization of double precision symmetric indefinite matrix to compute determinant and/or inverse. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SSIDI. | Class(es): D2b1a D3b1a | Usage: CALL DSIDI(A,LDA,N,KPVT,DET,INERT,WORK,JOB) | On-line doc: CALL GAMSDOC DSIDI (or @PRT CMLIB*DOC.DSIDI/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DSICO DSIFA
- DSIFA** Compute factorization of double precision symmetric indefinite matrix. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SSIFA. | Class(es): D2b1a | Usage: CALL DSIFA(A,LDA,N,KPVT,INFO) | On-line doc: CALL GAMSDOC DSIFA (or @PRT CMLIB*DOC.DSIFA/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB
- DSINDG** Double precision sine d in degrees. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is SINDG. | Class(es): C4a | Usage: D1 = DSINDG (D) | On-line doc: CALL GAMSDOC DSINDG (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DSINH** Computes hyperbolic sin sinh(x). | Proprietary double precision Fortran subprogram in PORT library. Single precision version is SINH. | Class(es): C4c | Usage: D = DSINH (X) | On-line doc: CALL GAMSDOC DSINH (or @PRT PORT*DOC.DSINH) | Access: LIB NBS*PORT
- DSISL** Use factorization of double precision symmetric indefinite matrix to solve systems. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SSISL. | Class(es): D2b1a | Usage: CALL DSISL(A,LDA,N,KPVT,B) | On-line doc: CALL GAMSDOC DSISL (or @PRT CMLIB*DOC.DSISL/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DSICO DSIFA
- DSMSNO** Minimize a general unconstrained objective function using finite difference gradients and secant Hessian approximations. (Double precision version of SMSNO.) | Portable double precision Fortran subprogram in DNL2SN sublibrary of CMLIB library. Single precision version is SMSNO. | Class(es): G1b1a | Usage: CALL DSMSNO(N,D,X,CALCF<IV,LIV,LV,V,UIPARM,URPARN,UFPARM) | On-line doc: CALL GAMSDOC DSMSNO (or @PRT CMLIB*DOC.DSMSNO/DNL2SN) | Tests: CMLIB*TEST-SOURCE.\$F2/DNL2SN, CMLIB*TEST-SOURCE.\$Q2/DNL2SN | Access: LIB NBS*CMLIB
- DSPCO** Compute factorization of double precision symmetric indefinite matrix stored in packed form and estimate its condition. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SSPCO. | Class(es): D2b1a | Usage: CALL DSPCO(AP,N,KPVT,RCOND,Z) | On-line doc: CALL GAMSDOC DSPCO (or @PRT CMLIB*DOC.DSPCO/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB

- DSPDI** Use factorization of double precision symmetric indefinite matrix stored in packed form to compute determinant and/or inverse. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SSPDI. | Class(es): D2b1a D3b1a | Usage: CALL DSPDI(AP,N,KPVT,DET,INERT,WORK,JOB) | On-line doc: CALL GAMSDOC DSPDI (or @PRT CMLIB*DOC.DSPDI/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DSPCO DSPFA
- DSPENC** Double precision Spence Dilogarithm, $= - \int_0^d \frac{(\ln(\text{abs.val. of } 1-y))}{y} dy$. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is SPENC. | Class(es): C5 | Usage: D1 = DSPENC (D) | On-line doc: CALL GAMSDOC DSPENC (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DSPFA** Compute factorization of double precision symmetric indefinite matrix stored in packed form. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SSPFA. | Class(es): D2b1a | Usage: CALL DSPFA(AP,N,KPVT,INFO) | On-line doc: CALL GAMSDOC DSPFA (or @PRT CMLIB*DOC.DSPFA/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB
- DSPLN1** Evaluates a function and derivatives described previously by an expansion in terms of B-splines. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is SPLN1. | Class(es): E3 K6 | Usage: CALL DSPLN1 (K,T,N,A,X,NX,ID,NID,FX) | On-line doc: CALL GAMSDOC DSPLN1 (or @PRT PORT*DOC.DSPLN1) | Access: LIB NBS*PORT
- DSPLN2** Evaluates a function described by a previously determined expansion in B-splines. More flexible than DSPLN1. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is SPLN2. | Class(es): E3 K6 | Usage: CALL DSPLN2 (K,T,N,A,X,NX,ID,NID,FX,IDIM,ADIFF,ILO,ILEFT) | On-line doc: CALL GAMSDOC DSPLN2 (or @PRT PORT*DOC.DSPLN2) | Access: LIB NBS*PORT
- DSPLND** Evaluates at a given set of points a function described by a previously determined expansion in terms of B-splines. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is SPLND. | Class(es): E3 K6 | Usage: CALL DSPLND (K,T,N,A,X,NX,MD,FX) | On-line doc: CALL GAMSDOC DSPLND (or @PRT PORT*DOC.DSPLND) | Access: LIB NBS*PORT
- DSPLNE** Evaluates at a set of points, a function described by a previously determined expansion in terms of B-splines. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is SPLNE. | Class(es): E3 K6 | Usage: CALL DSPLNE (K,T,N,A,X,NX,FX) | On-line doc: CALL GAMSDOC DSPLNE (or @PRT PORT*DOC.DSPLNE) | Access: LIB NBS*PORT
- DSPLNI** Integrates a function described previously by an expansion in terms of B-splines. Several integrations can be performed in one call. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is SPLNI. | Class(es): H2a2a1 E3 K6 | Usage: CALL DSPLNI (K,T,N,A,X,NX,FIX) | On-line doc: CALL GAMSDOC DSPLNI (or @PRT PORT*DOC.DSPLNI) | Access: LIB NBS*PORT
- DSPSL** Use factorization of double precision symmetric indefinite matrix stored in packed form to solve systems. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SSPSL. | Class(es): D2b1a | Usage: CALL DSPSL(AP,N,KPVT,B) | On-line doc: CALL GAMSDOC DSPSL (or @PRT CMLIB*DOC.DSPSL/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB | See also: DSPCO DSPFA
- DSUMSL** Minimizes a general unconstrained objective function using analytic gradient and a Hessian approximation from a secant update. (Double precision version of SUMSL.). | Portable double precision Fortran subprogram in DNL2SN sublibrary of CMLIB library. Single precision version is SUMSL. | Class(es): G1b1b | Usage: CALL DSUMSL(N,D,X,CALCF,GALCG,IV,LIV,LV,V,UIPARM,URPARM,UFPARM) | On-line doc: CALL GAMSDOC DSUMSL (or @PRT CMLIB*DOC.DSUMSL/DNL2SN) | Tests: CMLIB*TEST-SOURCE.\$F2/DNL2SN, CMLIB*TEST-SOURCE.\$Q2/DNL2SN | Access: LIB NBS*CMLIB
- DSVDC** Compute Singular Value Decomposition of double precision matrix. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SSVDC. | Class(es): D6 | Usage: CALL DSVDC(X,LDX,N,P,S,E,U,LDU,V,LDV,WORK,JOB,INFO) | On-line doc: CALL GAMSDOC DSVDC (or @PRT CMLIB*DOC.DSVDC/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB
- DSWAP** Interchange vectors X and Y, both double precision. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. Single precision version is SSWAP. | Class(es): D1a5 | Usage: CALL DSWAP(N,DX,INCX,DY,INCY) | On-line doc: CALL GAMSDOC DSWAP (or @PRT CMLIB*DOC.DSWAP/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- DTAN** Double precision tangent of d. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4a | Usage: D1 = DTAN (D) | On-line doc: CALL GAMSDOC DTAN (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- DTAN** Computes the elementary tangent function. If your Fortran library includes this function, use that instead. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is TAN. | Class(es): C4a | Usage: D = DTAN (X) | On-line doc: CALL GAMSDOC DTAN (or @PRT PORT*DOC.DTAN) | Access: LIB NBS*PORT
- DTANH** Computes hyperbolic tangent, $\tanh(x)$. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is TANH. | Class(es): C4c | Usage: D = DTANH (X) | On-line doc: CALL GAMSDOC DTANH (or @PRT PORT*DOC.DTANH) | Access: LIB NBS*PORT
- DTCHBP** Evaluates a polynomial expressed as a sum of Chebyshev polynomials. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is TCHBP. | Class(es): C3a2 | Usage: D = DTCHBP (N, ALPHA, X, X0, X1) | On-line doc: CALL GAMSDOC DTCHBP (or @PRT PORT*DOC.DTCHBP) | Access: LIB NBS*PORT
- DTPTB** Solve a system of ordinary differential equations with boundary conditions at two points, using a multiple shooting method. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): 11b2 | Usage: CALL DTPTB

(N,FCNI,FCNJ,FCNB,XA,XB,NITER,X,MAX,Y,IY,DTOL,BTOL, WORK,IER) | On-line doc: CALL GAMSDOC DTPTB (or @PRT IMSL*DOC.DTPTB) | Access: LIB NBS*IMSL

DTRCO Estimates condition of double precision triangular matrix. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is STRCO. | Class(es): D2a3 | Usage: CALL DTRCO(T,LDT,N,RCOND,Z,JOB) | On-line doc: CALL GAMSDOC DTRCO (or @PRT CMLIB*DOC.DTRCO/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB

DTRDI Computes determinant and/or inverse of double precision triangular matrix. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is STRDI. | Class(es): D2a3 D3a3 | Usage: CALL DTRDI(T,LDT,N,DET,JOB,INFO) | On-line doc: CALL GAMSDOC DTRDI (or @PRT CMLIB*DOC.DTRDI/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB

DTRIGP Evaluates a trigonometric polynomial with given coefficients. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is TRIGP. | Class(es): C3a1 | Usage: D = DTRIGP (N, ALPHA, BETA, THETA) | On-line doc: CALL GAMSDOC DTRIGP (or @PRT PORT*DOC.DTRIGP) | Access: LIB NBS*PORT

DTRSL Solves systems with double precision triangular matrix. | Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is STRSL. | Class(es): D2a3 | Usage: CALL DTRSL(T,LDT,N,B,JOB,INFO) | On-line doc: CALL GAMSDOC DTRSL (or @PRT CMLIB*DOC.DTRSL/LINPACKD) | Tests: CMLIB*TEST-SOURCE.\$F1/LINPACKD, CMLIB*TEST-SOURCE.\$F2/LINPACKD | Access: LIB NBS*CMLIB

DUMB Given interval endpoints, this generates a uniform mesh, with needed multiplicities for B-spline use. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is UMB. | Class(es): E8 K6 | Usage: CALL DUMB (A,B,NAB,K,X,NX) | On-line doc: CALL GAMSDOC DUMB (or @PRT PORT*DOC.DUMB) | Access: LIB NBS*PORT

DUMD Given interval endpoints, this generates a uniform mesh of distinct points. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is UMD. | Class(es): E8 K6 | Usage: CALL DUMD (A,B,NAB,X) | On-line doc: CALL GAMSDOC DUMD (or @PRT PORT*DOC.DUMD) | Access: LIB NBS*PORT

DUMKFL Decomposes a non-zero floating point number into a mantissa and an exponent. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is UMKFL. | Class(es): A6c | Usage: CALL DUMKFL (F, E, M) | On-line doc: CALL GAMSDOC DUMKFL (or @PRT PORT*DOC.DUMKFL) | Access: LIB NBS*PORT

DVBTD Converts a mantissa and exponent into a base 10 floating point number. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is VBTOD. | Class(es): A6b | Usage: CALL DVBTD (E, M, E10, M10) | On-line doc: CALL GAMSDOC DVBTD (or @PRT PORT*DOC.DVBTD) | Access: LIB NBS*PORT

DVCPR Solve a system of ordinary differential equations with boundary conditions at two points, using a variable order, variable step size finite difference method with deferred corrections. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): I1b2 | Usage: CALL DVCPR (N,FCNI,FCNJ,FCNB,XA,XB,NGMAX,NGRID,IP,IR,TOL,X,Y, IY,ABT,PAR, WORK, IWORK,IER) | On-line doc: CALL GAMSDOC DVCPR (or @PRT IMSL*DOC.DVCPR) | Access: LIB NBS*IMSL

DVDTOD Converts a base-10 mantissa and exponent of a floating point number into a machine-base representation. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): A6c | Usage: CALL DVDTOD (E10, M10, E, M) | On-line doc: CALL GAMSDOC DVDTOD (or @PRT PORT*DOC.DVDTOD) | Access: LIB NBS*PORT

DVERK Differential equation solver - Runge Kutta-Verner fifth and sixth order method. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): I1a1a | Usage: CALL DVERK (N,FCN,X,Y,XEND,TOL,IND,C,NW,W,IER) | On-line doc: CALL GAMSDOC DVERK (or @PRT IMSL*DOC.DVERK) | Access: LIB NBS*IMSL

DZERO Finds a single real root of a function within an interval specified by the user. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ZERO. | Class(es): F1b | Usage: D = DZERO (F,A,B,T) | On-line doc: CALL GAMSDOC DZERO (or @PRT PORT*DOC.DZERO) | Access: LIB NBS*PORT

DZONE Finds a solution of a system of non-linear equations. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ZONE. | Class(es): F2a | Usage: CALL DZONE (FUNC, N, X, EPS, JMAX, F2NORM) | On-line doc: CALL GAMSDOC DZONE (or @PRT PORT*DOC.DZONE) | Access: LIB NBS*PORT

DZONEJ Finds a solution of a system of non-linear equations. User must provide a SUBROUTINE to compute the Jacobian matrix. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ZONEJ. | Class(es): F2a | Usage: CALL DZONEJ (FUNC, DZIJAC, N, X, EPS, JMAX, F2NORM) | On-line doc: CALL GAMSDOC DZONEJ (or @PRT PORT*DOC.DZONEJ) | Access: LIB NBS*PORT

E

- E01AAE** Interpolated values, one variable, data at unequally spaced points, Aitken's technique. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E01AAF. | Class(es): E1b | Usage: CALL E01AAE (A, B, C, N1, N2, N, X) | On-line doc: CALL GAMSDOC E01AAE (or @PRT NAG*DOC.E01AAE) | Access: LIB NBS*NAG
- E01AAF** Interpolated values, one variable, data at unequally spaced points, Aitken's technique. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E01AAE. | Class(es): E1b | Usage: CALL E01AAF (A, B, C, N1, N2, N, X) | On-line doc: CALL GAMSDOC E01AAF (or @PRT NAG*DOC.E01AAF) | Access: LIB NBS*NAG
- E01ABE** Interpolated values, one variable, data at equally spaced points, Everett's formula. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E01ABF. | Class(es): E1b | Usage: CALL E01ABE (N, P, A, G, N1, N2, IFAIL) | On-line doc: CALL GAMSDOC E01ABE (or @PRT NAG*DOC.E01ABE) | Access: LIB NBS*NAG
- E01ABF** Interpolated values, one variable, data at equally spaced points, Everett's formula. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E01ABE. | Class(es): E1b | Usage: CALL E01ABF (N, P, A, G, N1, N2, IFAIL) | On-line doc: CALL GAMSDOC E01ABF (or @PRT NAG*DOC.E01ABF) | Access: LIB NBS*NAG
- E01ACE** Interpolated values, two variables, data on rectangular grid, fitting bicubic spline. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E01ACF. | Class(es): E2a | Usage: CALL E01ACE (A, B, X, Y, F, VAL, VALL, IFAIL, XX, WORK, AM, D, IG1, M1, N1) | On-line doc: CALL GAMSDOC E01ACE (or @PRT NAG*DOC.E01ACE) | Access: LIB NBS*NAG
- E01ACF** Interpolated values, two variables, data on rectangular grid, fitting bicubic spline. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E01ACE. | Class(es): E2a | Usage: CALL E01ACF (A, B, X, Y, F, VAL, VALL, IFAIL, XX, WORK, AM, D, IG1, M1, N1) | On-line doc: CALL GAMSDOC E01ACF (or @PRT NAG*DOC.E01ACF) | Access: LIB NBS*NAG
- E01AEE** Interpolating functions, polynomial interpolant, data may include derivative values. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E01AEF. | Class(es): E1b | Usage: CALL E01AEE (M, XMIN, XMAX, X, Y, IP, N, ITMIN, ITMAX, A, WRK, LWRK, IWRK, LIWRK IFAIL) | On-line doc: CALL GAMSDOC E01AEE (or @PRT NAG*DOC.E01AEE) | Access: LIB NBS*NAG | See also: E02AKE
- E01AEF** Interpolating functions, polynomial interpolant, data may include derivative values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E01AEE. | Class(es): E1b | Usage: CALL E01AEF (M, XMIN, XMAX, X, Y, IP, N, ITMIN, ITMAX, A, WRK, LWRK, IWRK, LIWRK IFAIL) | On-line doc: CALL GAMSDOC E01AEF (or @PRT NAG*DOC.E01AEF) | Access: LIB NBS*NAG | See also: E02AKF
- E01BAE** Interpolating functions cubic spline interpolant. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E01BAF. | Class(es): E1a | Usage: CALL E01BAE (M, X, Y, K, C, LCK, WRK, LWRK, IFAIL) | On-line doc: CALL GAMSDOC E01BAE (or @PRT NAG*DOC.E01BAE) | Access: LIB NBS*NAG | See also: E02BBE E02BCE E02BDE
- E01BAF** Interpolating functions cubic spline interpolant. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E01BAE. | Class(es): E1a | Usage: CALL E01BAF (M, X, Y, K, C, LCK, WRK, LWRK, IFAIL) | On-line doc: CALL GAMSDOC E01BAF (or @PRT NAG*DOC.E01BAF) | Access: LIB NBS*NAG | See also: E02BBF E02BCF E02BDF
- E01RAE** Produces, from a set of function values and corresponding abscissae, the coefficients of an interpolating rational function expressed in continued fraction form. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E01RAF. | Class(es): E1c | Usage: CALL E01RAE (N,X,F,M,A,U,IW,IFAIL) | On-line doc: CALL GAMSDOC E01RAE (or @PRT NAG*DOC.E01RAE) | Access: LIB NBS*NAG | See also: E01RBE
- E01RAF** Produces, from a set of function values and corresponding abscissae, the coefficients of an interpolating rational function expressed in continued fraction form. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E01RAE. | Class(es): E1c | Usage: CALL E01RAF (N,X,F,M,A,U,IW,IFAIL) | On-line doc: CALL GAMSDOC E01RAF (or @PRT NAG*DOC.E01RAF) | Access: LIB NBS*NAG | See also: E01RBF
- E01RBE** Evaluates continued fractions of the form produced by NAG library routine E01RAE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E01RBF. | Class(es): E3 | Usage: CALL E01RBE(M,A,U,X,F,IFAIL) | On-line doc: CALL GAMSDOC E01RBE (or @PRT NAG*DOC.E01RBE) | Access: LIB NBS*NAG
- E01RBF** Evaluates continued fractions of the form produced by NAG library routine E01RAF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E01RBE. | Class(es): E3 | Usage: CALL E01RBF(M,A,U,X,F,IFAIL) | On-line doc: CALL GAMSDOC E01RBF (or @PRT NAG*DOC.E01RBF) | Access: LIB NBS*NAG
- E02ACE** Minimax curve fit by polynomials. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02ACF. | Class(es): K2 | Usage: CALL E02ACE (X, Y, N, A, M1, REF) | On-line doc: CALL GAMSDOC E02ACE (or @PRT NAG*DOC.E02ACE) | Access: LIB NBS*NAG
- E02ACF** Minimax curve fit by polynomials. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02ACE. | Class(es): K2 | Usage: CALL E02ACF (X, Y, N, A, M1, REF) | On-line doc: CALL GAMSDOC E02ACF (or @PRT NAG*DOC.E02ACF) | Access: LIB NBS*NAG
- E02ADE** Least-squares curve fit by polynomials, arbitrary data points. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02ADF. | Class(es): K1a1a2 | Usage: CALL E02ADE (M, KPLUS1, NROWS, X, Y, W, WORK1, WORK2, A, S, IFAIL) | On-line doc: CALL GAMSDOC E02ADE (or @PRT NAG*DOC.E02ADE) | Access: LIB NBS*NAG | See also: E02AEE

- E02ADF** Least-squares curve fit by polynomials, arbitrary data points. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02ADE. | Class(es): K1a1a2 | Usage: CALL E02ADF (M, KPLUS1, NROWS, X, Y, W, WORK1, WORK2, A, S, IFAIL) | On-line doc: CALL GAMSDOC E02ADF (or @PRT NAG*DOC.E02ADF) | Access: LIB NBS*NAG | See also: E02AEF
- E02AEE** Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form (simplified parameter list). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02AEF. | Class(es): C3a2 E3 K6 | Usage: CALL E02AEE (NPLUS1, A, XCAP, P, IFAIL) | On-line doc: CALL GAMSDOC E02AEE (or @PRT NAG*DOC.E02AEE) | Access: LIB NBS*NAG
- E02AEF** Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form (simplified parameter list). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02AEE. | Class(es): C3a2 E3 K6 | Usage: CALL E02AEF (NPLUS1, A, XCAP, P, IFAIL) | On-line doc: CALL GAMSDOC E02AEF (or @PRT NAG*DOC.E02AEF) | Access: LIB NBS*NAG
- E02AFE** Least-squares curve fit by polynomials, special data points (including interpolation). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02AFF. | Class(es): K1a1a2 E1b | Usage: CALL E02AFE (NPLUS1, F, A, IFAIL) | On-line doc: CALL GAMSDOC E02AFE (or @PRT NAG*DOC.E02AFE) | Access: LIB NBS*NAG | See also: E02AEE
- E02AFF** Least-squares curve fit by polynomials, special data points (including interpolation). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02AFE. | Class(es): K1a1a2 E1b | Usage: CALL E02AFF (NPLUS1, F, A, IFAIL) | On-line doc: CALL GAMSDOC E02AFF (or @PRT NAG*DOC.E02AFF) | Access: LIB NBS*NAG | See also: E02AEF
- E02AGE** Least-squares curve fit by polynomials, arbitrary data points, values and derivatives may be constrained. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02AGF. | Class(es): K1a2a | Usage: CALL E02AGE (M, KPLUS1, NROWS, XMIN, XMAX, X, Y, W, MF, XF, YF, LYF, IP, A, S, NP1, WRK, LWRK, IWRK, LIWRK, IFAIL) | On-line doc: CALL GAMSDOC E02AGE (or @PRT NAG*DOC.E02AGE) | Access: LIB NBS*NAG | See also: E02AKE
- E02AGF** Least-squares curve fit by polynomials, arbitrary data points, values and derivatives may be constrained. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02AGE. | Class(es): K1a2a | Usage: CALL E02AGF (M, KPLUS1, NROWS, XMIN, XMAX, X, Y, W, MF, XF, YF, LYF, IP, A, S, NP1, WRK, LWRK, IWRK, LIWRK, IFAIL) | On-line doc: CALL GAMSDOC E02AGF (or @PRT NAG*DOC.E02AGF) | Access: LIB NBS*NAG | See also: E02AKF
- E02AHE** Derivative of fitted polynomial in Chebyshev series form. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02AHF. | Class(es): C3a2 E3 K6 | Usage: CALL E02AHE (NP1, XMIN, XMAX, A, IA1, LA, PATM1, ADIF, IADIF1, LADIF, IFAIL) | On-line doc: CALL GAMSDOC E02AHE (or @PRT NAG*DOC.E02AHE) | Access: LIB NBS*NAG
- E02AHF** Derivative of fitted polynomial in Chebyshev series form. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02AHE. | Class(es): C3a2 E3 K6 | Usage: CALL E02AHF (NP1, XMIN, XMAX, A, IA1, LA, PATM1, ADIF, IADIF1, LADIF, IFAIL) | On-line doc: CALL GAMSDOC E02AHF (or @PRT NAG*DOC.E02AHF) | Access: LIB NBS*NAG
- E02AJE** Integral of fitted polynomial in Chebyshev series form. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02AJF. | Class(es): H2a2a1 C3a2 K6 | Usage: CALL E02AJE (NP1, XMIN, XMAX, A, IA1, LA, QATM1, AINT, IAIN1, LAINT, IFAIL) | On-line doc: CALL GAMSDOC E02AJE (or @PRT NAG*DOC.E02AJE) | Access: LIB NBS*NAG
- E02AJF** Integral of fitted polynomial in Chebyshev series form. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02AJE. | Class(es): H2a2a1 C3a2 K6 | Usage: CALL E02AJF (NP1, XMIN, XMAX, A, IA1, LA, QATM1, AINT, IAIN1, LAINT, IFAIL) | On-line doc: CALL GAMSDOC E02AJF (or @PRT NAG*DOC.E02AJF) | Access: LIB NBS*NAG
- E02AKE** Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02AKF. | Class(es): C3a2 E3 K6 | Usage: CALL E02AKE (NP1, XMIN, XMAX, A, IA1, LA, X, RESULT, IFAIL) | On-line doc: CALL GAMSDOC E02AKE (or @PRT NAG*DOC.E02AKE) | Access: LIB NBS*NAG
- E02AKF** Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02AKE. | Class(es): C3a2 E3 K6 | Usage: CALL E02AKF (NP1, XMIN, XMAX, A, IA1, LA, X, RESULT, IFAIL) | On-line doc: CALL GAMSDOC E02AKF (or @PRT NAG*DOC.E02AKF) | Access: LIB NBS*NAG
- E02BAE** Least-squares curve fit by cubic splines (including interpolation). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02BAF. | Class(es): K1a1a1 E1a | Usage: CALL E02BAE (M, NCAP7, X, Y, W, K, WORK1, WORK2, C, SS, IFAIL) | On-line doc: CALL GAMSDOC E02BAE (or @PRT NAG*DOC.E02BAE) | Access: LIB NBS*NAG | See also: E02BBE E02BCE E02BDE
- E02BAF** Least-squares curve fit by cubic splines (including interpolation). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02BAE. | Class(es): K1a1a1 E1a | Usage: CALL E02BAF (M, NCAP7, X, Y, W, K, WORK1, WORK2, C, SS, IFAIL) | On-line doc: CALL GAMSDOC E02BAF (or @PRT NAG*DOC.E02BAF) | Access: LIB NBS*NAG | See also: E02BBF E02BCF E02BDF
- E02BBE** Evaluation of fitted functions, cubic spline as E02BAE, function only. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02BBF. | Class(es): E3 K6 | Usage: CALL E02BBE (NCAP7, K, C, X, S, IFAIL) | On-line doc: CALL GAMSDOC E02BBE (or @PRT NAG*DOC.E02BBE) | Access: LIB NBS*NAG
- E02BBF** Evaluation of fitted functions, cubic spline as E02BAF, function only. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02BBE. | Class(es): E3 K6 | Usage: CALL E02BBF (NCAP7, K, C, X, S, IFAIL) | On-line doc: CALL GAMSDOC E02BBF (or @PRT NAG*DOC.E02BBF) | Access: LIB NBS*NAG
- E02BCE** Evaluation of fitted functions, cubic spline as E02BAE, function and derivatives. | Proprietary single precision Fortran subprogram

in NAG library. Double precision version is E02BCF. | Class(es): E3 K6 | Usage: CALL E02BCE (NCAP7, K, C, X, LEFT, S, IFAIL) |
On-line doc: CALL GAMSDOC E02BCE (or @PRT NAG*DOC.E02BCE) | Access: LIB NBS*NAG

E02BCF Evaluation of fitted functions, cubic spline as E02BAF, function and derivatives. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02BCE. | Class(es): E3 K6 | Usage: CALL E02BCF (NCAP7, K, C, X, LEFT, S, IFAIL) |
On-line doc: CALL GAMSDOC E02BCF (or @PRT NAG*DOC.E02BCF) | Access: LIB NBS*NAG

E02BDE Evaluation of fitted functions, cubic spline as E02BAE, definite integral. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02BDF. | Class(es): H2a2a1 E3 K6 | Usage: CALL E02BDE (NCAP7, K, C, DEFINIT, IFAIL) |
On-line doc: CALL GAMSDOC E02BDE (or @PRT NAG*DOC.E02BDE) | Access: LIB NBS*NAG

E02BDF Evaluation of fitted functions, cubic spline as E02BAF, definite integral. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02BDE. | Class(es): H2a2a1 E3 K6 | Usage: CALL E02BDF (NCAP7, K, C, DEFINIT, IFAIL) |
On-line doc: CALL GAMSDOC E02BDF (or @PRT NAG*DOC.E02BDF) | Access: LIB NBS*NAG

E02CAE Least-squares surface fit by polynomials, for data on lines. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02CAF. | Class(es): K1a1b | Usage: CALL E02CAE (M, N, K, L, X, Y, F, W, NX, A, NA, XMIN, XMAX, NUX, INUXP1, NUY, INUYPI, WORK, NWORK, IFAIL) | On-line doc: CALL GAMSDOC E02CAE (or @PRT NAG*DOC.E02CAE) |
Access: LIB NBS*NAG

E02CAF Least-squares surface fit by polynomials, for data on lines. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02CAE. | Class(es): K1a1b | Usage: CALL E02CAF (M, N, K, L, X, Y, F, W, NX, A, NA, XMIN, XMAX, NUX, INUXP1, NUY, INUYPI, WORK, NWORK, IFAIL) | On-line doc: CALL GAMSDOC E02CAF (or @PRT NAG*DOC.E02CAF) |
Access: LIB NBS*NAG

E02CBE Evaluation of fitted functions, polynomial in two variables as E02CAE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02CBF. | Class(es): E3 K6 | Usage: CALL E02CBE (MFIRST, MLAST, K, L, X, XMIN, XMAX, Y, YMIN, YMAX, FF, A, NA, WORK, NWORK, IFAIL) | On-line doc: CALL GAMSDOC E02CBE (or @PRT NAG*DOC.E02CBE) |
Access: LIB NBS*NAG

E02CBF Evaluation of fitted functions, polynomial in two variables as E02CAF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02CBE. | Class(es): E3 K6 | Usage: CALL E02CBF (MFIRST, MLAST, K, L, X, XMIN, XMAX, Y, YMIN, YMAX, FF, A, NA, WORK, NWORK, IFAIL) | On-line doc: CALL GAMSDOC E02CBF (or @PRT NAG*DOC.E02CBF) |
Access: LIB NBS*NAG

E02DAE Least-squares surface fit by bicubic splines. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02DAF. | Class(es): K1a1b | Usage: CALL E02DAE (M, PX, PY, X, Y, F, W, LAMDA, MU, POINT, NPOINT, DL, C, NC, WS, NWS EPS, SIGMA, RANK, IFAIL) | On-line doc: CALL GAMSDOC E02DAE (or @PRT NAG*DOC.E02DAE) | Access: LIB NBS*NAG | See also: E02DBE

E02DAF Least-squares surface fit by bicubic splines. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02DAE. | Class(es): K1a1b | Usage: CALL E02DAF (M, PX, PY, X, Y, F, W, LAMDA, MU, POINT, NPOINT, DL, C, NC, WS, NWS EPS, SIGMA, RANK, IFAIL) | On-line doc: CALL GAMSDOC E02DAF (or @PRT NAG*DOC.E02DAF) | Access: LIB NBS*NAG | See also: E02DBF

E02DBE Evaluation of fitted functions, bicubic spline as E02DAE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02DBF. | Class(es): E3 K6 | Usage: CALL E02DBE (M, PX, PY, X, Y, FF, LAMDA, MU, POINT, NPOINT, C, NC, IFAIL) | On-line doc: CALL GAMSDOC E02DBE (or @PRT NAG*DOC.E02DBE) | Access: LIB NBS*NAG

E02DBF Evaluation of fitted functions, bicubic spline as E02DAF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02DBE. | Class(es): E3 K6 | Usage: CALL E02DBF (M, PX, PY, X, Y, FF, LAMDA, MU, POINT, NPOINT, C, NC, IFAIL) | On-line doc: CALL GAMSDOC E02DBF (or @PRT NAG*DOC.E02DBF) | Access: LIB NBS*NAG

E02GAE L1-approximation by general linear function. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02GAF. | Class(es): D9 K3 | Usage: CALL E02GAE (M, A, LA, B, NPLUS2, TOLER, X, RESID, IRANK, ITER, IWORK, IFAIL) | On-line doc: CALL GAMSDOC E02GAE (or @PRT NAG*DOC.E02GAE) | Access: LIB NBS*NAG

E02GAF L1-approximation by general linear function. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02GAE. | Class(es): D9 K3 | Usage: CALL E02GAF (M, A, LA, B, NPLUS2, TOLER, X, RESID, IRANK, ITER, IWORK, IFAIL) | On-line doc: CALL GAMSDOC E02GAF (or @PRT NAG*DOC.E02GAF) | Access: LIB NBS*NAG

E02GBE L1-approximation by general linear function subject to linear inequality constraints. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02GBF. | Class(es): D9 K3 | Usage: CALL E02GBE (M, N, MPL, E, IE, F, X, MXS, MONIT, IPRINT, K, ELIN, INDX, W, IW, IFAIL) | On-line doc: CALL GAMSDOC E02GBE (or @PRT NAG*DOC.E02GBE) |
Access: LIB NBS*NAG

E02GBF L1-approximation by general linear function subject to linear inequality constraints. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02GBE. | Class(es): D9 K3 | Usage: CALL E02GBF (M, N, MPL, E, IE, F, X, MXS, MONIT, IPRINT, K, ELIN, INDX, W, IW, IFAIL) | On-line doc: CALL GAMSDOC E02GBF (or @PRT NAG*DOC.E02GBF) |
Access: LIB NBS*NAG

E02GCE Calculates an L-infinity solution to an over-determined system of linear equations. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02GCF. | Class(es): D9 K2 | Usage: CALL E02GCE (M, N, MDIM, NDIM, A, B, TOL, RELERR, X, RESMAX, IRANK, ITER, IFAIL) | On-line doc: CALL GAMSDOC E02GCE (or @PRT NAG*DOC.E02GCE) | Access: LIB NBS*NAG

- E02GCF** Calculates an L-infinity solution to an over-determined system of linear equations. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02GCE. | Class(es): D9 K2 | Usage: CALL E02GCF (M, N, MDIM, NDIM, A, B, TOL, RELERR, X, RESMAX, IRANK, ITER, IFAIL) | On-line doc: CALL GAMSDOC E02GCF (or @PRT NAG*DOC.E02GCF) | Access: LIB NBS*NAG
- E02RAE** Pade-approximants. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02RAF. | Class(es): K4 | Usage: CALL E02RAE (IA, IB, C, IC, A, B, W, JW, IFAIL) | On-line doc: CALL GAMSDOC E02RAE (or @PRT NAG*DOC.E02RAE) | Access: LIB NBS*NAG | See also: E02RBE
- E02RAF** Pade-approximants. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02RAE. | Class(es): K4 | Usage: CALL E02RAF (IA, IB, C, IC, A, B, W, JW, IFAIL) | On-line doc: CALL GAMSDOC E02RAF (or @PRT NAG*DOC.E02RAF) | Access: LIB NBS*NAG | See also: E02RBF
- E02RBE** Evaluation of fitted functions, rational function as E02RAE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02RBF. | Class(es): K6 | Usage: CALL E02RBE (A, IA, B, IB, X, ANS, IFAIL) | On-line doc: CALL GAMSDOC E02RBE (or @PRT NAG*DOC.E02RBE) | Access: LIB NBS*NAG
- E02RBF** Evaluation of fitted functions, rational function as E02RAF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02RBE. | Class(es): K6 | Usage: CALL E02RBF (A, IA, B, IB, X, ANS, IFAIL) | On-line doc: CALL GAMSDOC E02RBF (or @PRT NAG*DOC.E02RBF) | Access: LIB NBS*NAG
- E02ZAE** Sort 2-d data into panels for fitting or evaluating bicubic splines. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02ZAF. | Class(es): E3 K6 | Usage: CALL E02ZAE (PX, PY, LAMDA, MU, M, X, Y, POINT, NPOINT, ADRES, NADRES, IFAIL) | On-line doc: CALL GAMSDOC E02ZAE (or @PRT NAG*DOC.E02ZAE) | Access: LIB NBS*NAG
- E02ZAF** Sort 2-d data into panels for fitting or evaluating bicubic splines. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02ZAE. | Class(es): E3 K6 | Usage: CALL E02ZAF (PX, PY, LAMDA, MU, M, X, Y, POINT, NPOINT, ADRES, NADRES, IFAIL) | On-line doc: CALL GAMSDOC E02ZAF (or @PRT NAG*DOC.E02ZAF) | Access: LIB NBS*NAG
- E04ABE** Minimum, function of one variable using function values only. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04ABF. | Class(es): G1a1a | Usage: CALL E04ABE (FUNCT, E1, E2, A, B, MAXCAL, X, F, IFAIL) | On-line doc: CALL GAMSDOC E04ABE (or @PRT NAG*DOC.E04ABE) | Access: LIB NBS*NAG
- E04ABF** Minimum, function of one variable using function values only. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04ABE. | Class(es): G1a1a | Usage: CALL E04ABF (FUNCT, E1, E2, A, B, MAXCAL, X, F, IFAIL) | On-line doc: CALL GAMSDOC E04ABF (or @PRT NAG*DOC.E04ABF) | Access: LIB NBS*NAG
- E04BBE** Minimum, function of one variable, using first derivative. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04BBF. | Class(es): G1a1b | Usage: CALL E04BBE (FUNCT, E1, E2, A, B, MAXCAL, X, F, G, IFAIL) | On-line doc: CALL GAMSDOC E04BBE (or @PRT NAG*DOC.E04BBE) | Access: LIB NBS*NAG
- E04BBF** Minimum, function of one variable, using first derivative. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04BBE. | Class(es): G1a1b | Usage: CALL E04BBF (FUNCT, E1, E2, A, B, MAXCAL, X, F, G, IFAIL) | On-line doc: CALL GAMSDOC E04BBF (or @PRT NAG*DOC.E04BBF) | Access: LIB NBS*NAG
- E04CCE** Unconstrained minimum, function of several variables (comprehensive), using function values only, simplex algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04CCF. | Class(es): G1b2 | Usage: CALL E04CCE(N, X, F, TOL, IW, W1, W2, W3, W4, W5, W6, FUNCT, MONIT, MAXCAL, IFAIL) | On-line doc: CALL GAMSDOC E04CCE (or @PRT NAG*DOC.E04CCE) | Access: LIB NBS*NAG
- E04CCF** Unconstrained minimum, function of several variables (comprehensive), using function values only, simplex algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04CCE. | Class(es): G1b2 | Usage: CALL E04CCF(N, X, F, TOL, IW, W1, W2, W3, W4, W5, W6, FUNCT, MONIT, MAXCAL, IFAIL) | On-line doc: CALL GAMSDOC E04CCF (or @PRT NAG*DOC.E04CCF) | Access: LIB NBS*NAG
- E04CGE** Unconstrained minimum, function of several variables (easy-to-use), using function values only, quasi-Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04CGF. | Class(es): G1b1a | Usage: CALL E04CGE (N, X, F, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04CGE (or @PRT NAG*DOC.E04CGE) | Access: LIB NBS*NAG
- E04CGF** Unconstrained minimum, function of several variables (easy-to-use), using function values only, quasi-Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04CGE. | Class(es): G1b1a | Usage: CALL E04CGF (N, X, F, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04CGF (or @PRT NAG*DOC.E04CGF) | Access: LIB NBS*NAG
- E04DBE** Unconstrained minimum, function of several variables (comprehensive), using first derivatives, conjugate direction algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04DBF. | Class(es): G1b1b | Usage: CALL E04DBE (N, X, F, G, XTOL, FEST, DUM, W, FUNCT, MONIT, MAXCAL, IFAIL) | On-line doc: CALL GAMSDOC E04DBE (or @PRT NAG*DOC.E04DBE) | Access: LIB NBS*NAG
- E04DBF** Unconstrained minimum, function of several variables (comprehensive), using first derivatives, conjugate direction algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04DBE. | Class(es): G1b1b | Usage: CALL E04DBF (N, X, F, G, XTOL, FEST, DUM, W, FUNCT, MONIT, MAXCAL, IFAIL) | On-line doc: CALL GAMSDOC E04DBF (or @PRT NAG*DOC.E04DBF) | Access: LIB NBS*NAG

- E04DEE** Unconstrained minimum, function of several variables (easy-to-use), using first derivatives, quasi-Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04DEF. | Class(es): G1b1b | Usage: CALL E04DEE (N, X, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04DEE (or @PRT NAG*DOC.E04DEE) | Access: LIB NBS*NAG
- E04DEF** Unconstrained minimum, function of several variables (easy-to-use), using first derivatives, quasi-Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04DEE. | Class(es): G1b1b | Usage: CALL E04DEF (N, X, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04DEF (or @PRT NAG*DOC.E04DEF) | Access: LIB NBS*NAG
- E04DFE** Unconstrained minimum, function of several variables (easy-to-use), using first derivatives, modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04DFF. | Class(es): G1b1b | Usage: CALL E04DFE (N, X, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04DFE (or @PRT NAG*DOC.E04DFE) | Access: LIB NBS*NAG
- E04DFF** Unconstrained minimum, function of several variables (easy-to-use), using first derivatives, modified Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04DFE. | Class(es): G1b1b | Usage: CALL E04DFF (N, X, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04DFF (or @PRT NAG*DOC.E04DFF) | Access: LIB NBS*NAG
- E04EBE** Unconstrained minimum, function of several variables (easy-to-use), using first and second derivatives, modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04EBF. | Class(es): G1b1c | Usage: CALL E04EBE (N, X, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04EBE (or @PRT NAG*DOC.E04EBE) | Access: LIB NBS*NAG
- E04EBF** Unconstrained minimum, function of several variables (easy-to-use), using first and second derivatives, modified Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04EBE. | Class(es): G1b1c | Usage: CALL E04EBF (N, X, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04EBF (or @PRT NAG*DOC.E04EBF) | Access: LIB NBS*NAG
- E04FCE** Unconstrained minimum, sum of squares, n variables (comprehensive), using function values only, combined Gauss-Newton and modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04FCF. | Class(es): K1b1a1 | Usage: CALL E04FCE (M, N, LSQFUN, LSQMON, IPRINT, MAXCAL, ETA, XTOL, STEPMX, X, FSUMSQ, FVEC, FJAC, LJ, S, V, LV, NITER, NF, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04FCE (or @PRT NAG*DOC.E04FCE) | Access: LIB NBS*NAG
- E04FCF** Unconstrained minimum, sum of squares, n variables (comprehensive), using function values only, combined Gauss-Newton and modified Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04FCE. | Class(es): K1b1a1 | Usage: CALL E04FCF (M, N, LSQFUN, LSQMON, IPRINT, MAXCAL, ETA, XTOL, STEPMX, X, FSUMSQ, FVEC, FJAC, LJ, S, V, LV, NITER, NF, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04FCF (or @PRT NAG*DOC.E04FCF) | Access: LIB NBS*NAG
- E04FDE** Unconstrained minimum, sum of squares, n variables (easy-to-use), function values only, Gauss-Newton and modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04FDF. | Class(es): K1b1a1 | Usage: CALL E04FDE (M, N, X, FSUMSQ, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04FDE (or @PRT NAG*DOC.E04FDE) | Access: LIB NBS*NAG
- E04FDF** Unconstrained minimum, sum of squares, n variables (easy-to-use), function values only, Gauss-Newton and modified Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04FDE. | Class(es): K1b1a1 | Usage: CALL E04FDF (M, N, X, FSUMSQ, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04FDF (or @PRT NAG*DOC.E04FDF) | Access: LIB NBS*NAG
- E04GBE** Unconstrained minimum, sum of squares, n variables (comprehensive), using first derivatives, Gauss-Newton and quasi-Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04GBF. | Class(es): K1b1a2 | Usage: CALL E04GBE (M, N, LSQLIN, LSQFUN, LSQMON, IPRINT, MAXCAL, ETA, XTOL, STEPMX, X, FSUMSQ, FVEC, FJAC, LJ, S, V, LV, NITER, NF, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04GBE (or @PRT NAG*DOC.E04GBE) | Access: LIB NBS*NAG
- E04GBF** Unconstrained minimum, sum of squares, n variables (comprehensive), using first derivatives, Gauss-Newton and quasi-Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04GBE. | Class(es): K1b1a2 | Usage: CALL E04GBF (M, N, LSQLIN, LSQFUN, LSQMON, IPRINT, MAXCAL, ETA, XTOL, STEPMX, X, FSUMSQ, FVEC, FJAC, LJ, S, V, LV, NITER, NF, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04GBF (or @PRT NAG*DOC.E04GBF) | Access: LIB NBS*NAG
- E04GCE** Unconstrained minimum, sum of squares, n variables (easy-to-use), using first derivatives, Gauss-Newton and quasi-Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04GCF. | Class(es): K1b1a2 | Usage: CALL E04GCE (M, N, X, FSUMSQ, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04GCE (or @PRT NAG*DOC.E04GCE) | Access: LIB NBS*NAG
- E04GCF** Unconstrained minimum, sum of squares, n variables (easy-to-use), using first derivatives, Gauss-Newton and quasi-Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04GCE. | Class(es): K1b1a2 | Usage: CALL E04GCF (M, N, X, FSUMSQ, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04GCF (or @PRT NAG*DOC.E04GCF) | Access: LIB NBS*NAG
- E04GDE** Unconstrained minimum, sum of squares, n variables (comprehensive), using first derivatives, Gauss-Newton and modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04GDF. | Class(es): K1b1a2 | Usage: CALL E04GDE (M, N, LSQMON, IPRINT, MAXCAL, ETA, XTOL, STEPMX, X, FSUMSQ, FVEC, FJAC, LJ, S, V, LV, NITER, NF, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04GDE (or @PRT NAG*DOC.E04GDE) | Access: LIB NBS*NAG
- E04GDF** Unconstrained minimum, sum of squares, n variables (comprehensive), using first derivatives, Gauss-Newton and modified Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04GDE. | Class(es): K1b1a2 |

Usage: CALL E04GDF (M, N, LSQMON, IPRINT, MAXCAL, ETA, XTOL, STEPMX, X, FSUMSQ, FVEC, FJAC, LJ, S, V, LV, NITER, NF, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04GDF (or @PRT NAG*DOC.E04GDF) | Access: LIB NBS*NAG

E04GEE Unconstrained minimum, sum of squares, n variables (easy-to-use), using first derivatives, Gauss-Newton and modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04GEF. | Class(es): K1b1a2 | Usage: CALL E04GEE (M, N, X, FSUMSQ, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04GEE (or @PRT NAG*DOC.E04GEE) | Access: LIB NBS*NAG

E04GEF Unconstrained minimum, sum of squares, n variables (easy-to-use), using first derivatives, Gauss-Newton and modified Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04GEE. | Class(es): K1b1a2 | Usage: CALL E04GEF (M, N, X, FSUMSQ, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04GEF (or @PRT NAG*DOC.E04GEF) | Access: LIB NBS*NAG

E04HBE Finite-difference intervals for estimating first derivatives. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04HBF. | Class(es): G4f | Usage: CALL E04HBE (N, FUNCT, X, NF, DELTA, HESL, LH, HESD, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04HBE (or @PRT NAG*DOC.E04HBE) | Access: LIB NBS*NAG | See also: E04JBE

E04HBF Finite-difference intervals for estimating first derivatives. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04HBE. | Class(es): G4f | Usage: CALL E04HBF (N, FUNCT, X, NF, DELTA, HESL, LH, HESD, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04HBF (or @PRT NAG*DOC.E04HBF) | Access: LIB NBS*NAG | See also: E04JBF

E04HCE Check user's routine calculating first derivatives of function. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04HCF. | Class(es): G4c F3 | Usage: CALL E04HCE (N, FUNCT, X, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04HCE (or @PRT NAG*DOC.E04HCE) | Access: LIB NBS*NAG

E04HCF Check user's routine calculating first derivatives of function. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04HCE. | Class(es): G4c F3 | Usage: CALL E04HCF (N, FUNCT, X, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04HCF (or @PRT NAG*DOC.E04HCF) | Access: LIB NBS*NAG

E04HDE Check user's routine calculating second derivatives of function. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04HDF. | Class(es): G4c F3 | Usage: CALL E04HDE (N, FUNCT, HESS, X, G, HESL, LH, HESD, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04HDE (or @PRT NAG*DOC.E04HDE) | Access: LIB NBS*NAG

E04HDF Check user's routine calculating second derivatives of function. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04HDE. | Class(es): G4c F3 | Usage: CALL E04HDF (N, FUNCT, HESS, X, G, HESL, LH, HESD, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04HDF (or @PRT NAG*DOC.E04HDF) | Access: LIB NBS*NAG

E04HEE Unconstrained minimum, sum of squares, n variables (comprehensive), using second derivatives, Gauss-Newton and modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04HEF. | Class(es): K1b1a3 | Usage: CALL E04HEE (M, N, LSQFUN, LSQHESS, LSQMON, IPRINT, MAXCAL, ETA, XTOL, STEPMS, X, FSUMSQ, FVEC, FJAC, LJ, S, V, LV, NITER, NF, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04HEE (or @PRT NAG*DOC.E04HEE) | Access: LIB NBS*NAG

E04HEF Unconstrained minimum, sum of squares, n variables (comprehensive), using second derivatives, Gauss-Newton and modified Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04HEE. | Class(es): K1b1a3 | Usage: CALL E04HEF (M, N, LSQFUN, LSQHESS, LSQMON, IPRINT, MAXCAL, ETA, XTOL, STEPMS, X, FSUMSQ, FVEC, FJAC, LJ, S, V, LV, NITER, NF, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04HEF (or @PRT NAG*DOC.E04HEF) | Access: LIB NBS*NAG

E04HFE Unconstrained minimum, sum of squares, n variables (easy-to-use), using second derivatives, Gauss-Newton and modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04HFF. | Class(es): K1b1a3 | Usage: CALL E04HFE (M, N, X, FSUMSQ, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04HFE (or @PRT NAG*DOC.E04HFE) | Access: LIB NBS*NAG

E04HFF Unconstrained minimum, sum of squares, n variables (easy-to-use), using second derivatives, Gauss-Newton and modified Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04HFE. | Class(es): K1b1a3 | Usage: CALL E04HFF (M, N, X, FSUMSQ, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04HFF (or @PRT NAG*DOC.E04HFF) | Access: LIB NBS*NAG

E04JAE Minimum, function of several variables, simple bounds (easy-to-use), using function values only, quasi-Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04JAF. | Class(es): G2h1a1 | Usage: CALL E04JAE (N, IBOUND, BL, BU, X, F, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04JAE (or @PRT NAG*DOC.E04JAE) | Access: LIB NBS*NAG

E04JAF Minimum, function of several variables, simple bounds (easy-to-use), using function values only, quasi-Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04JAE. | Class(es): G2h1a1 | Usage: CALL E04JAF (N, IBOUND, BL, BU, X, F, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04JAF (or @PRT NAG*DOC.E04JAF) | Access: LIB NBS*NAG

E04JBE Minimum, function of several variables, simple bounds (comprehensive), using function values only, quasi-Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04JBF. | Class(es): G2h1a1 G1b1a | Usage: CALL E04JBE (N, FUNCT, MONIT, IPRINT, LOCSH, INTYPE, MINLIN, MAXCAL, ETA, XTOL, STEPMX, FEST, DELTA, IBOUND, BL, BU, X, HESL, LH, HESD, ISTATE, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04JBE (or @PRT NAG*DOC.E04JBE) | Access: LIB NBS*NAG

E04JBF Minimum, function of several variables, simple bounds (comprehensive), using function values only, quasi-Newton algorithm. |

Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04JBE. | Class(es): G2h1a1 G1b1a | Usage: CALL E04JBF (N, FUNCT, MONIT, IPRINT, LOC SH, INTYPE, MINLIN, MAXCAL, ETA, XTOL, STEPMX, FEST, DELTA, IBOUND, BL, BU, X, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04JBF (or @PRT NAG*DOC.E04JBF) | Access: LIB NBS*NAG

E04KAE Minimum, function of several variables, simple bounds (easy-to-use), using first derivatives, quasi-Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04KAF. | Class(es): G2h1a2 | Usage: CALL E04KAE (N, IBOUND, BL, BU, X, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04KAE (or @PRT NAG*DOC.E04KAE) | Access: LIB NBS*NAG

E04KAF Minimum, function of several variables, simple bounds (easy-to-use), using first derivatives, quasi-Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04KAE. | Class(es): G2h1a2 | Usage: CALL E04KAF (N, IBOUND, BL, BU, X, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04KAF (or @PRT NAG*DOC.E04KAF) | Access: LIB NBS*NAG

E04KBE Minimum, function of several variables, simple bounds (comprehensive), using first derivatives, quasi-Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04KBF. | Class(es): G2h1a2 | Usage: CALL E04KBE (N, FUNCT, MONIT, IPRINT, LOC SCH, INTYPE, MINLIN, MAXCAL, ETA, XTOL, STEPMX, FEST, IBOUND, BL, BU, X, HESL, LH, HESD, ISTATE, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04KBE (or @PRT NAG*DOC.E04KBE) | Access: LIB NBS*NAG

E04KBF Minimum, function of several variables, simple bounds (comprehensive), using first derivatives, quasi-Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04KBE. | Class(es): G2h1a2 | Usage: CALL E04KBF (N, FUNCT, MONIT, IPRINT, LOC SCH, INTYPE, MINLIN, MAXCAL, ETA, XTOL, STEPMX, FEST, IBOUND, BL, BU, X, HESL, LH, HESD, ISTATE, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04KBF (or @PRT NAG*DOC.E04KBF) | Access: LIB NBS*NAG

E04KCE Minimum, function of several variables, simple bounds (easy-to-use), using first derivatives, modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04KCF. | Class(es): G2h1a2 | Usage: CALL E04KCE (N, IBOUND, BL, BU, X, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04KCE (or @PRT NAG*DOC.E04KCE) | Access: LIB NBS*NAG

E04KCF Minimum, function of several variables, simple bounds (easy-to-use), using first derivatives, modified Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04KCE. | Class(es): G2h1a2 | Usage: CALL E04KCF (N, IBOUND, BL, BU, X, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04KCF (or @PRT NAG*DOC.E04KCF) | Access: LIB NBS*NAG

E04KDE Minimum, function of several variables, simple bounds (comprehensive), using first derivatives, modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04KDF. | Class(es): G2h1a2 | Usage: CALL E04KDE (N, FUNCT, MONIT, IPRINT, MAXCAL, ETA, XTOL, DELTA, STEPMX, IBOUND, BL, BU, X, HESL, LH, HESD, ISTATE, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04KDE (or @PRT NAG*DOC.E04KDE) | Access: LIB NBS*NAG

E04KDF Minimum, function of several variables, simple bounds (comprehensive), using first derivatives, modified Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04KDE. | Class(es): G2h1a2 | Usage: CALL E04KDF (N, FUNCT, MONIT, IPRINT, MAXCAL, ETA, XTOL, DELTA, STEPMX, IBOUND, BL, BU, X, HESL, LH, HESD, ISTATE, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04KDF (or @PRT NAG*DOC.E04KDF) | Access: LIB NBS*NAG

E04LAE Minimum, function of several variables, simple bounds (easy-to-use), using first and second derivatives, modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04LAF. | Class(es): G2h1a3 | Usage: CALL E04LAE (N, IBOUND, BL, BU, X, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04LAE (or @PRT NAG*DOC.E04LAE) | Access: LIB NBS*NAG

E04LAF Minimum, function of several variables, simple bounds (easy-to-use), using first and second derivatives, modified Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04LAE. | Class(es): G2h1a3 | Usage: CALL E04LAF (N, IBOUND, BL, BU, X, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04LAF (or @PRT NAG*DOC.E04LAF) | Access: LIB NBS*NAG

E04LBE Minimum, function of several variables, simple bounds (comprehensive), using first and second derivatives, modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04LBF. | Class(es): G2h1a3 | Usage: CALL E04LBE (N, FUNCT, HESS, MONIT, IPRINT, MAXCAL, ETA, XTOL, STEPMX, IBOUND, BL, BU, X, HESL, LH, HESD, ISTATE, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04LBE (or @PRT NAG*DOC.E04LBE) | Access: LIB NBS*NAG

E04LBF Minimum, function of several variables, simple bounds (comprehensive), using first and second derivatives, modified Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04LBE. | Class(es): G2h1a3 | Usage: CALL E04LBF (N, FUNCT, HESS, MONIT, IPRINT, MAXCAL, ETA, XTOL, STEPMX, IBOUND, BL, BU, X, HESL, LH, HESD, ISTATE, F, G, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04LBF (or @PRT NAG*DOC.E04LBF) | Access: LIB NBS*NAG

E04UAE Minimum, function of n variables, non-linear constraints, function and constraint values only, sequential augmented Lagrangian quasi-Newton method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04UAF. | Class(es): G2h3b1a | Usage: CALL E04UAE (N, MEQ, MINEQ, MRNGE, M, MONAUX, IPRINT, MAXCAL, ETA, XTOL, STEPMX, CL, CU, LCLU, IBOUND, XL, XU, LAMSET, X, RHO, RLAM, F, C, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04UAE (or @PRT NAG*DOC.E04UAE) | Access: LIB NBS*NAG

E04UAF Minimum, function of n variables, non-linear constraints, function and constraint values only, sequential augmented Lagrangian

quasi-Newton method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04UAE. | Class(es): G2h3b1a | Usage: CALL E04UAF (N, MEQ, MINEQ, MRNGE, M, MONAUX, IPRINT, MAXCAL, ETA, XTOL, STEPMX, CL, CU, LCLU, IBOUND, XL, XU, LAMSET, X, RHO, RLAM, F, G, C, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04UAF (or @PRT NAG*DOC.E04UAF) | Access: LIB NBS*NAG

E04VAE Minimum, function of several variables, general non-linear constraints, using first derivatives, sequential augmented Lagrangian quasi-Newton method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04VAF. | Class(es): G2h3b1b | Usage: CALL E04VAE (N, MEQ, MINEQ, MRNGE, M, MONAUX, IPRINT, SALMIN, MAXCAL, ETA, XTOL, STEPMX, CL, CU, LCLU, IBOUND, XL, XU, LAMSET, X, RHO, RLAM, F, G, C, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04VAE (or @PRT NAG*DOC.E04VAE) | Access: LIB NBS*NAG

E04VAF Minimum, function of several variables, general non-linear constraints, using first derivatives, sequential augmented Lagrangian quasi-Newton method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04VAE. | Class(es): G2h3b1b | Usage: CALL E04VAF (N, MEQ, MINEQ, MRNGE, M, MONAUX, IPRINT, SALMIN, MAXCAL, ETA, XTOL, STEPMX, CL, CU, LCLU, IBOUND, XL, XU, LAMSET, X, RHO, RLAM, F, G, C, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04VAF (or @PRT NAG*DOC.E04VAF) | Access: LIB NBS*NAG

E04VBE Minimum, function of several variables, general non-linear constraints, using first derivatives, sequential augmented Lagrangian modified Newton method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04VBF. | Class(es): G2h3b1b | Usage: CALL E04VBE (N, MEQ, MINEQ, MRNGE, M, IPRINT, MAXCAL, ETA, XTOL, STEPMX, CL, CU, LCLU, IBOUND, XL, XU, LAMSET, X, RHO, RLAM, F, G, C, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04VBE (or @PRT NAG*DOC.E04VBE) | Access: LIB NBS*NAG

E04VBF Minimum, function of several variables, general non-linear constraints, using first derivatives, sequential augmented Lagrangian modified Newton method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04VBE. | Class(es): G2h3b1b | Usage: CALL E04VBF (N, MEQ, MINEQ, MRNGE, M, IPRINT, MAXCAL, ETA, XTOL, STEPMX, CL, CU, LCLU, IBOUND, XL, XU, LAMSET, X, RHO, RLAM, F, G, C, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04VBF (or @PRT NAG*DOC.E04VBF) | Access: LIB NBS*NAG

E04WAE Minimum, function of n variables, non-linear constraints, first and second derivatives, sequential augmented Lagrangian modified Newton method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04WAF. | Class(es): G2h3b1c | Usage: CALL E04WAE (N, MEQ, MINEQ, MRNGE, M, IPRINT, MAXCAL, ETA, XTOL, STEPMX, CL, CU, LCLU, IBOUND, XL, XU, LAMSET, X, RHO, RLAM, F, G, C, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04WAE (or @PRT NAG*DOC.E04WAE) | Access: LIB NBS*NAG

E04WAF Minimum, function of n variables, non-linear constraints, first and second derivatives, sequential augmented Lagrangian modified Newton method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04WAE. | Class(es): G2h3b1c | Usage: CALL E04WAF (N, MEQ, MINEQ, MRNGE, M, IPRINT, MAXCAL, ETA, XTOL, STEPMX, CL, CU, LCLU, IBOUND, XL, XU, LAMSET, X, RHO, RLAM, F, G, C, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04WAF (or @PRT NAG*DOC.E04WAF) | Access: LIB NBS*NAG

E04YAE Check user's routine calculating Jacobian matrix of first derivatives. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04YAF. | Class(es): G4c | Usage: CALL E04YAE (M, N, LSQFUN, X, FVEC, FJAC, LJ, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04YAE (or @PRT NAG*DOC.E04YAE) | Access: LIB NBS*NAG

E04YAF Check user's routine calculating Jacobian matrix of first derivatives. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04YAE. | Class(es): G4c | Usage: CALL E04YAF (M, N, LSQFUN, X, FVEC, FJAC, LJ, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04YAF (or @PRT NAG*DOC.E04YAF) | Access: LIB NBS*NAG

E04YBE Check user's routine calculating second derivative term in Hessian matrix of sum of squares. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04YBF. | Class(es): G4c | Usage: CALL E04YBE (M, N, LSQFUN, LSQHES, X, FVEC, FJAC, LJ, B, LB, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04YBE (or @PRT NAG*DOC.E04YBE) | Access: LIB NBS*NAG

E04YBF Check user's routine calculating second derivative term in Hessian matrix of sum of squares. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04YBE. | Class(es): G4c | Usage: CALL E04YBF (M, N, LSQFUN, LSQHES, X, FVEC, FJAC, LJ, B, LB, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04YBF (or @PRT NAG*DOC.E04YBF) | Access: LIB NBS*NAG

E04ZAE Check user's routines calculating first derivatives of function and constraints. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04ZAF. | Class(es): G4c | Usage: CALL E04ZAE (N, M, FUNCT, CON, X, F, G, C, A, LA, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04ZAE (or @PRT NAG*DOC.E04ZAE) | Access: LIB NBS*NAG

E04ZAF Check user's routines calculating first derivatives of function and constraints. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04ZAE. | Class(es): G4c | Usage: CALL E04ZAF (N, M, FUNCT, CON, X, F, G, C, A, LA, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04ZAF (or @PRT NAG*DOC.E04ZAF) | Access: LIB NBS*NAG

E04ZBE Check user's routines calculating second derivatives of function and constraints. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04ZBF. | Class(es): G4c | Usage: CALL E04ZBE (N, M, FUNCT, HESS, CON, CHESS, X, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04ZBE (or @PRT NAG*DOC.E04ZBE) | Access: LIB NBS*NAG

E04ZBF Check user's routines calculating second derivatives of function and constraints. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04ZBE. | Class(es): G4c | Usage: CALL E04ZBF (N, M, FUNCT, HESS, CON, CHESS, X, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04ZBF (or @PRT NAG*DOC.E04ZBF) | Access: LIB NBS*NAG

- E1** Exponential integral, the integral from x to infinity of $(e^{*-t} / t) dt$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DE1. | Class(es): C5 | Usage: $R = E1(X)$ | On-line doc: CALL GAMSDOC E1 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- EBALAC** Balance a complex general matrix and isolate eigenvalues whenever possible. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4c1a | Usage: CALL EBALAC (AR,A1,N,IA,K,L,D) | On-line doc: CALL GAMSDOC EBALAC (or @PRT IMSL*DOC.EBALAC) | Access: LIB NBS*IMSL
- EBALAF** Balance a real matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4c1a | Usage: CALL EBALAF (A,N,IA,D,K,L) | On-line doc: CALL GAMSDOC EBALAF (or @PRT IMSL*DOC.EBALAF) | Access: LIB NBS*IMSL
- EEBSF** Estimates the error in a given B-spline fit to a function, f , by refining the mesh. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DEBSF. | Class(es): E3 K6 | Usage: $X = EEBSF(K,T1,N1,A1,T2,N2,A2)$ | On-line doc: CALL GAMSDOC EEBSF (or @PRT PORT*DOC.EEBSF) | Access: LIB NBS*PORT
- EEBSI** Estimates the error in a given B-spline fit to a function f by refining the mesh intervals selected by user. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DEBSI. | Class(es): E3 K6 | Usage: $X = EEBSI(K,T1,N1,A1,T2,N2,A2,X,NX,EREST1,EREST2)$ | On-line doc: CALL GAMSDOC EEBSI (or @PRT PORT*DOC.EEBSI) | Access: LIB NBS*PORT
- EESFF** Finds the maximum absolute error in a given B-spline fit to a function, f . | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DEESFF. | Class(es): E3 K6 | Usage: $X = EESFF(K,T,N,A,F)$ | On-line doc: CALL GAMSDOC EESFF (or @PRT PORT*DOC.EESFF) | Access: LIB NBS*PORT
- EESFI** Finds the maximum absolute error in a given B-spline fit to a function, f , on a set of user selected intervals. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DEESFI. | Class(es): E3 K6 | Usage: $X = EESFI(K,T,N,A,F,X,NX,EEST)$ | On-line doc: CALL GAMSDOC EESFI (or @PRT PORT*DOC.EESFI) | Access: LIB NBS*PORT
- EHESSC** Reduction of a general complex matrix to complex upper Hessenberg form. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4c1b2 | Usage: CALL EHESSC (AR,A1,K,L,N,IA,ID) | On-line doc: CALL GAMSDOC EHESSC (or @PRT IMSL*DOC.EHESSC) | Access: LIB NBS*IMSL
- EHESSF** Reduction of a nonsymmetric matrix to upper Hessenberg form by orthogonal transformations. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4c1b2 | Usage: CALL EHESSF (A,K,L,N,IA,D) | On-line doc: CALL GAMSDOC EHESSF (or @PRT IMSL*DOC.EHESSF) | Access: LIB NBS*IMSL
- EHOUH** Reduction of a complex Hermitian matrix to real symmetric tridiagonal form. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4c1b1 | Usage: CALL EHOUSH (AR,A1,N,D,E,TAU) | On-line doc: CALL GAMSDOC EHOUSH (or @PRT IMSL*DOC.EHOUSH) | Access: LIB NBS*IMSL
- EHOUS** Reduction of a symmetric matrix to symmetric tridiagonal form using a Householder reduction. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4c1b1 | Usage: CALL EHOUS (A,N,D,E,E2) | On-line doc: CALL GAMSDOC EHOUS (or @PRT IMSL*DOC.EHOUS) | Access: LIB NBS*IMSL
- EI** Exponential integral, = the integral from $-x$ to infinity of $(e^{*-t} / t) dt$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DEI. | Class(es): C5 | Usage: $R = EI(X)$ | On-line doc: CALL GAMSDOC EI (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- EIGBS** Find some eigenvalues and (optionally) eigenvectors of a real symmetric band matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4a6 | Usage: CALL EIGBS (A,N,IA,IJOB,NC,M,D,Z,IZ,WORK,IER) | On-line doc: CALL GAMSDOC EIGBS (or @PRT IMSL*DOC.EIGBS) | Access: LIB NBS*IMSL
- EIGCC** Eigenvalues and (optionally) eigenvectors of a complex general matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4a4 | Usage: CALL EIGCC (A,N,IA,IJOB,W,Z,IZ,WK,IER) | On-line doc: CALL GAMSDOC EIGCC (or @PRT IMSL*DOC.EIGCC) | Access: LIB NBS*IMSL
- EIGCH** Eigenvalues and (optionally) eigenvectors of a complex Hermitian matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4a3 | Usage: CALL EIGCH (A,N,IJOB,D,Z,IZ,WK,IER) | On-line doc: CALL GAMSDOC EIGCH (or @PRT IMSL*DOC.EIGCH) | Access: LIB NBS*IMSL
- EIGEN** Finds all eigenvalues and eigenvectors of a real matrix. Output consists of pairs of real arrays. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DEIGEN. | Class(es): D4a2 | Usage: CALL EIGEN (NM,N,A,WR,WI,Z) | On-line doc: CALL GAMSDOC EIGEN (or @PRT PORT*DOC.EIGEN) | Access: LIB NBS*PORT
- EIGRF** Eigenvalues and (optionally) eigenvectors of a real general matrix in full storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4a2 | Usage: CALL EIGRF (A,N,IA,IJOB,W,Z,IZ,WK,IER) | On-line doc: CALL GAMSDOC EIGRF (or @PRT IMSL*DOC.EIGRF) | Access: LIB NBS*IMSL
- EIGRS** Eigenvalues and (optionally) eigenvectors of a real symmetric matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4a1 | Usage: CALL EIGRS (A,N,IJOB,D,Z,IZ,WK,IER) | On-line doc: CALL GAMSDOC EIGRS (or @PRT IMSL*DOC.EIGRS) | Access: LIB NBS*IMSL
- EIGZC** Eigenvalues and (optionally) eigenvectors of the system $A*x = \lambda B*x$ where A and B are complex matrices. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4b4 | Usage: CALL EIGZC

(A,IA,B,IB,N,IJOB,EIGA,EIGB,Z,IZ,WK,INFER,IER) | On-line doc: CALL GAMSDOC EIGZC (or @PRT IMSL*DOC.EIGZC) | Access: LIB NBS*IMSL

EIGZF Eigenvalues and (optionally) eigenvectors of the system $A*x = \lambda*B*x$ where A and B are real matrices. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4b2 | Usage: CALL EIGZF (A,IA,B,IB,N,IJOB,ALFA,BETA,Z,IZ,WK,IER) | On-line doc: CALL GAMSDOC EIGZF (or @PRT IMSL*DOC.EIGZF) | Access: LIB NBS*IMSL

EIGZS Eigenvalues and (optionally) eigenvectors of the system $A*x = \lambda*B*x$ where A and B are real symmetric matrices and B is positive definite. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4b1 | Usage: CALL EIGZS (A,B,N,IJOB,D,Z,IZ,WK,IER) | On-line doc: CALL GAMSDOC EIGZS (or @PRT IMSL*DOC.EIGZS) | Access: LIB NBS*IMSL

ELMBAK Forms eigenvectors of real general matrix from eigenvectors of upper Hessenberg matrix output from ELMHES. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL ELMBAK(NM,LOW,IGH,A,INT,M,Z) | On-line doc: CALL GAMSDOC ELMBAK (or @PRT CMLIB*DOC.ELMBAK/EISPACK) | Access: LIB NBS*CMLIB | See also: ELMHES

ELMHES Reduces real general matrix to upper Hessenberg form using stabilized elementary similarity transformations. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1b2 | Usage: CALL ELMHES(NM,N,LOW,IGH,A,INT) | On-line doc: CALL GAMSDOC ELMHES (or @PRT CMLIB*DOC.ELMHES/EISPACK) | Access: LIB NBS*CMLIB

ELTRAN Accumulates the stabilized elementary similarity transformations used in the reduction of a real general matrix to upper Hessenberg form by ELMHES. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL ELTRAN(NM,N,LOW,IGH,A,INT,Z) | On-line doc: CALL GAMSDOC ELTRAN (or @PRT CMLIB*DOC.ELTRAN/EISPACK) | Access: LIB NBS*CMLIB | See also: ELMHES

ENTER Save current error recovery mode and storage allocation status for PORT library programs. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): R3c | Usage: CALL ENTER (IRNEW) | On-line doc: CALL GAMSDOC ENTER (or @PRT PORT*DOC.ENTER) | Access: LIB NBS*PORT

ENTSRC Saves current recovery mode status and sets a new one for PORT library programs. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): R3a | Usage: CALL ENTSRC (IOLD,IRNEW) | On-line doc: CALL GAMSDOC ENTSRC (or @PRT PORT*DOC.ENTSRC) | Access: LIB NBS*PORT

EPRINT Print the current error message if the program is in the error state for PORT library programs. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): R3c | Usage: CALL EPRINT | On-line doc: CALL GAMSDOC EPRINT (or @PRT PORT*DOC.EPRINT) | Access: LIB NBS*PORT

EQRT1S Smallest or largest m eigenvalues of a symmetric tridiagonal matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4a5 | Usage: CALL EQRT1S (D,E2,N,M,ISW,IER) | On-line doc: CALL GAMSDOC EQRT1S (or @PRT IMSL*DOC.EQRT1S) | Access: LIB NBS*IMSL

EQRT2S Eigenvalues and (optionally) eigenvectors of a symmetric tridiagonal matrix using the QL method. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4a5 | Usage: CALL EQRT2S (D,E,N,Z,IZ,IER) | On-line doc: CALL GAMSDOC EQRT2S (or @PRT IMSL*DOC.EQRT2S) | Access: LIB NBS*IMSL

EQRT3S The smallest (or largest) eigenvalues of a tridiagonal matrix in algebraic value whose sum exceeds a given value. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4a5 | Usage: CALL EQRT3S (D,E2,N,VALUE,M,ISW,INFER,IER) | On-line doc: CALL GAMSDOC EQRT3S (or @PRT IMSL*DOC.EQRT3S) | Access: LIB NBS*IMSL

ERF Error function, $= (2 / \text{square root of pi}) * \text{the integral from 0 to x of } e^{-(t^2/2)} dt$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DERF. | Class(es): C8a L5a1e | Usage: R = ERF (X) | On-line doc: CALL GAMSDOC ERF (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

ERF Evaluate the error function. Note: the Fortran mathematical subroutine libraries may also contain ERF. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): C8a L5a1e | Usage: X = ERF (Y) | On-line doc: CALL GAMSDOC ERF (or @PRT IMSL*DOC.ERF) | Access: LIB NBS*IMSL

ERFC Complementary error function, $= (2 / \text{square root of pi}) * \text{the integral from x to infinity of } e^{-(t^2/2)} dt$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DERFC. | Class(es): C8a L5a1e | Usage: R = ERFC (X) | On-line doc: CALL GAMSDOC ERFC (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

ERFC Evaluate the complemented error function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): C8a L5a1e | Usage: X = ERFC (Y) | On-line doc: CALL GAMSDOC ERFC (or @PRT IMSL*DOC.ERFC) | Access: LIB NBS*IMSL

ERRINT Computes error function and complementary error function to maximum machine accuracy. To change computers change one line. | Portable double precision Fortran subprogram in STEGUN sublibrary of MATHWARE library. | Class(es): C8a L5a1e | Usage: CALL ERRINT(X,ERF,ERFC) | On-line doc: @PRT,S MATHWARE*STEGUN.ERRINT/DOC | Access: See individual sublibrary documentation

ERROFF Turns off the error state for PORT library programs. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): R3c | Usage: CALL ERROFF | On-line doc: CALL GAMSDOC ERROFF (or @PRT PORT*DOC.ERROFF) | Access: LIB NBS*PORT

EV1CDF Computes the cumulative distribution function value for the extreme value type 1 distribution. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1e | Usage: CALL EV1CDF(X,CDF) | On-line doc: CALL GAMSDOC EV1CDF (or @PRT DATAPAC*DOC.EV1CDF) | Access: LIB NBS*DAPAC

- EV1PLT** Generates an extreme value type 1 probability plot with mean = Euler's number = 0.57721566 and standard deviation = $\pi/\sqrt{6}$.
| Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c4e | Usage: CALL EV1PLT(X,N) | On-line doc: CALL GAMSDOC EV1PLT (or @PRT DATAPAC*DOC.EV1PLT) | Access: LIB NBS*DATAPAC
- EV1PPF** Computes the percent point function value for the extreme value type 1 distribution with mean = Euler's number = 0.57721566.
| Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2e | Usage: CALL EV1PPF(P,PPF) | On-line doc: CALL GAMSDOC EV1PPF (or @PRT DATAPAC*DOC.EV1PPF) | Access: LIB NBS*DATAPAC
- EV1RAN** Generates a random sample of size N from the extreme value type 1 distribution with mean = Euler's number = 0.57721566.
| Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a5 | Usage: CALL EV1RAN(N,ISTART,X) | On-line doc: CALL GAMSDOC EV1RAN (or @PRT DATAPAC*DOC.EV1RAN) | Access: LIB NBS*DATAPAC
- EV2CDF** Computes the cumulative distribution function value for the extreme value type 2 distribution with tail length parameter = GAMMA.
| Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1e | Usage: CALL EV2CDF(X,GAMMA,CDF) | On-line doc: CALL GAMSDOC EV2CDF (or @PRT DATAPAC*DOC.EV2CDF) | Access: LIB NBS*DATAPAC
- EV2PLT** Generates an extreme value type 2 probability plot with tail length parameter = GAMMA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c4e | Usage: CALL EV2PLT(X,N,GAMMA) | On-line doc: CALL GAMSDOC EV2PLT (or @PRT DATAPAC*DOC.EV2PLT) | Access: LIB NBS*DATAPAC
- EV2PPF** Computes the percent point function value for the extreme value type 2 distribution with tail length parameter = GAMMA.
| Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2e | Usage: CALL EV2PPF(P,GAMMA,PPF) | On-line doc: CALL GAMSDOC EV2PPF (or @PRT DATAPAC*DOC.EV2PPF) | Access: LIB NBS*DATAPAC
- EV2RAN** Generates a random sample of size N from the extreme value type 2 distribution with tail length parameter = GAMMA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a5 | Usage: CALL EV2RAN(N,GAMMA,ISTART,X) | On-line doc: CALL GAMSDOC EV2RAN (or @PRT DATAPAC*DOC.EV2RAN) | Access: LIB NBS*DATAPAC
- EXINT** Computes sequences of exponential integrals $E(N+K,X)$ $K=0,\dots,M-1$ or $\exp(X)$ times same to specified tolerance. | Portable single precision Fortran subprogram in AMOSLIB sublibrary of CMLIB library. Double precision version is DEXINT. | Class(es): C5 | Usage: CALL EXINT(X,N,KODE,TOL,EN,IERR) | On-line doc: CALL GAMSDOC EXINT (or @PRT CMLIB*DOC.EXINT/AMOSLIB) | Tests: CMLIB*TEST-SOURCE.EXINT/AMOSLIB | Access: LIB NBS*CMLIB
- EXP** Exponential function, $= e^{**x}$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4b | Usage: R = EXP (X) | On-line doc: CALL GAMSDOC EXP (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- EXPCDF** Computes the cumulative distribution function value for the exponential distribution with mean = 1 and standard deviation = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1e | Usage: CALL EXPCDF(X,CDF) | On-line doc: CALL GAMSDOC EXPCDF (or @PRT DATAPAC*DOC.EXPCDF) | Access: LIB NBS*DATAPAC
- EXPINT** Computes exponential integral $E_{\text{sub } n}(x)$. Change computer by changing one line. | Portable double precision Fortran subprogram in STEGUN sublibrary of MATHWARE library. | Class(es): C5 | Usage: CALL EXPINT(RN,X,ENX,EXPENX,IERR) | On-line doc: @PRT,S MATHWARE*STEGUN.EXPINT/DOC | Access: See individual sublibrary documentation
- EXPPDF** Computes the probability density function value for the exponential distribution with mean = 1 and standard deviation = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1e | Usage: CALL EXPPDF(X,N) | On-line doc: CALL GAMSDOC EXPPDF (or @PRT DATAPAC*DOC.EXPPDF) | Access: LIB NBS*DATAPAC
- EXPPLT** Generates an exponential probability plot with mean = 1 and standard deviation = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c4e | Usage: CALL EXPPLT(X,N) | On-line doc: CALL GAMSDOC EXPPLT (or @PRT DATAPAC*DOC.EXPPLT) | Access: LIB NBS*DATAPAC
- EXPPPF** Computes the percent point function value for the exponential distribution with mean = 1 and standard deviation = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2e | Usage: CALL EXPPPF(P,PPF) | On-line doc: CALL GAMSDOC EXPPPF (or @PRT DATAPAC*DOC.EXPPPF) | Access: LIB NBS*DATAPAC
- EXPRAN** Generates a random sample of size N from the exponential distribution with mean = 1 and standard deviation = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a5 | Usage: CALL EXPRAN(N,ISTART,X) | On-line doc: CALL GAMSDOC EXPRAN (or @PRT DATAPAC*DOC.EXPRAN) | Access: LIB NBS*DATAPAC
- EXPREL** Relative error exponential from first order, $= ((e^{**x}) - 1) / x$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DEXREL. | Class(es): C4b | Usage: R = EXPREL (X) | On-line doc: CALL GAMSDOC EXPREL (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- EXPSF** Computes the sparsity function value for the exponential distribution with mean = 1 and standard deviation = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2e | Usage: CALL EXPSF(P,SF) | On-line doc: CALL GAMSDOC EXPSF (or @PRT DATAPAC*DOC.EXPSF) | Access: LIB NBS*DATAPAC
- EXTREM** Performs an extreme value analysis on the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L4a1a5 | Usage: CALL EXTREM(X,N) | On-line doc: CALL GAMSDOC EXTREM (or @PRT DATAPAC*DOC.EXTREM) | Access: LIB NBS*DATAPAC
- EXTRMD** Finds extreme points of a double precision function defined on a mesh. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is EXTRMR. | Class(es): N5a D1a2 | Usage: CALL EXTRMD (NPPTS, FN, NEX, IEXT, IMAX, IMIN, IMAG) | On-line doc: CALL GAMSDOC EXTRMD (or @PRT PORT*DOC.EXTRMD) | Access: LIB NBS*PORT

- EXTRMI** Finds extremal points of an integer function defined on a mesh. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): N5a D1a2 | Usage: CALL EXTRMI (NPTS, FN, NEX, IEXT, IMAX, IMIN, IMAG) | On-line doc: CALL GAMSDOC EXTRMI (or @PRT PORT*DOC.EXTRMI) | Access: LIB NBS*PORT
- EXTRMR** Finds extremal points of a real function defined on a mesh. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is EXTRMD. | Class(es): N5a D1a2 | Usage: CALL EXTRMR (NPTS, FN, NEX, IEXT, IMAX, IMIN, IMAG) | On-line doc: CALL GAMSDOC EXTRMR (or @PRT PORT*DOC.EXTRMR) | Access: LIB NBS*PORT
- EZFFTB** Backward real discrete (fast) Fourier transform. Performs Fourier synthesis. Easy to use version. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): J1a1 | Usage: CALL EZFFTB(N,R,AZERO,A,B,WSAVE) | On-line doc: CALL GAMSDOC EZFFTB (or @PRT CMLIB*DOC.EZFFTB/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: EZFFTF
- EZFFTF** Forward real discrete (fast) Fourier transform. Performs Fourier analysis. Easy to use version. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): J1a1 | Usage: CALL EZFFTF(N,R,AZERO,A,B,WSAVE) | On-line doc: CALL GAMSDOC EZFFTF (or @PRT CMLIB*DOC.EZFFTF/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: EZFFTB

F

- F01AAE** Calculates the approximate inverse of a real matrix by Crout's method with partial pivoting. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AAF. | Class(es): D2a1 | Usage: CALL F01AAE(A, IA, N, UNIT, IUNIT, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F01AAE (or @PRT NAG*DOC.F01AAE) | Access: LIB NBS*NAG
- F01AAF** Calculates the approximate inverse of a real matrix by Crout's method with partial pivoting. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AAE. | Class(es): D2a1 | Usage: CALL F01AAF (A, IA, N, UNIT, IUNIT, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F01AAF (or @PRT NAG*DOC.F01AAF) | Access: LIB NBS*NAG
- F01ABE** Calculates the accurate inverse of a real symmetric positive definite matrix by Cholesky's method and iterative refinement. (Simplified parameter list). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01ABF. | Class(es): D2b1b | Usage: CALL F01ABE (A, IA, N, B, IB, Z, IFAIL) | On-line doc: CALL GAMSDOC F01ABE (or @PRT NAG*DOC.F01ABE) | Access: LIB NBS*NAG
- F01ABF** Calculates the accurate inverse of a real symmetric positive definite matrix by Cholesky's method and iterative refinement. (Simplified parameter list.). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01ABE. | Class(es): D2b1b | Usage: CALL F01ABF (A, IA, N, B, IB, Z, IFAIL) | On-line doc: CALL GAMSDOC F01ABF (or @PRT NAG*DOC.F01ABF) | Access: LIB NBS*NAG
- F01ACE** Calculates the accurate inverse of a real symmetric positive definite matrix by Cholesky's method and iterative refinement. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01ACF. | Class(es): D2b1b | Usage: CALL F01ACE (N, EPS, A, IA, B, IB, Z, L, IFAIL) | On-line doc: CALL GAMSDOC F01ACE (or @PRT NAG*DOC.F01ACE) | Access: LIB NBS*NAG
- F01ACF** Calculates the accurate inverse of a real symmetric positive definite matrix by Cholesky's method and iterative refinement. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01ACE. | Class(es): D2b1b | Usage: CALL F01ACF (N, EPS, A, IA, B, IB, Z, L, IFAIL) | On-line doc: CALL GAMSDOC F01ACF (or @PRT NAG*DOC.F01ACF) | Access: LIB NBS*NAG
- F01ADE** F01ADE calculates the approximate inverse of a real symmetric positive definite matrix by Cholesky's method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01ADF. | Class(es): D2b1b | Usage: CALL F01ADE (N, A, IA, IFAIL) | On-line doc: CALL GAMSDOC F01ADE (or @PRT NAG*DOC.F01ADE) | Access: LIB NBS*NAG
- F01ADF** Calculates the approximate inverse of a real symmetric positive definite matrix by Cholesky's method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01ADE. | Class(es): D2b1b | Usage: CALL F01ADF (N, A, IA, IFAIL) | On-line doc: CALL GAMSDOC F01ADF (or @PRT NAG*DOC.F01ADF) | Access: LIB NBS*NAG
- F01AEE** Reduces the generalized eigenproblem $Ax = (\lambda)Bx$, where A is real symmetric and B is real symmetric positive definite, to the standard symmetric eigenproblem. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AEF. | Class(es): D4c1c | Usage: CALL F01AEE (N, A, IA, B, IB, DL, IFAIL) | On-line doc: CALL GAMSDOC F01AEE (or @PRT NAG*DOC.F01AEE) | Access: LIB NBS*NAG
- F01AEF** Reduces the generalised eigenproblem $A*x=\lambda*B*x$ to the standard symmetric eigenproblem $P*z=\lambda*B*z$, A is a real symmetric matrix, B is a real symmetric positive definite matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AEE. | Class(es): D4c1c | Usage: CALL F01AEF (N, A, IA, B, IB, DL, IFAIL) | On-line doc: CALL GAMSDOC F01AEF (or @PRT NAG*DOC.F01AEF) | Access: LIB NBS*NAG
- F01AFE** Derives eigenvectors of several generalized eigenproblems from the corresponding eigenvectors of the related standard symmetric eigenproblem. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AFF. | Class(es): D4c4 | Usage: CALL F01AFE (N, M1, M2, B, IB, DL, Z, IZ) | On-line doc: CALL GAMSDOC F01AFE (or @PRT NAG*DOC.F01AFE) | Access: LIB NBS*NAG
- F01AFF** Derives eigenvectors x of the eigenproblems $A*x=\lambda*B*x$, $A*k*B*x=\lambda*B*x$ $y*B*a=\lambda*B*a$, (y is transpose of x), from the corresponding eigenvectors $z=M*x$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AFE. | Class(es): D4c4 | Usage: CALL F01AFF (N, M1, M2, B, IB, DL, Z, IZ) | On-line doc: CALL GAMSDOC F01AFF (or @PRT NAG*DOC.F01AFF) | Access: LIB NBS*NAG
- F01AGE** Gives the Householder reduction of a real symmetric matrix to tridiagonal form for use in F02BEF, F02AVF and F02BFF. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AGF. | Class(es): D4c1b1 | Usage: CALL F01AGE (N, TOL, A, IA, D, E, E2) | On-line doc: CALL GAMSDOC F01AGE (or @PRT NAG*DOC.F01AGE) | Access: LIB NBS*NAG | See also: F02BEF, F02AVF, F02BFF
- F01AGF** Gives the Householder reduction of a real symmetric matrix to tridiagonal form for use in F02BEF, F02AVF and F02BFF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AGE. | Class(es): D4c1b1 | Usage: CALL F01AGF (N, TOL, A, IA, D, E, E2) | On-line doc: CALL GAMSDOC F01AGF (or @PRT NAG*DOC.F01AGF) | Access: LIB NBS*NAG | See also: F02BEF, F02AVF, F02BFF
- F01AHE** Derives eigenvectors of a real symmetric matrix from the eigenvectors of the tridiagonal form produced by F01AGE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AHF. | Class(es): D4c4 | Usage: CALL F01AHE (N, M1, M2, A, IA, E, Z, IZ) | On-line doc: CALL GAMSDOC F01AHE (or @PRT NAG*DOC.F01AHE) | Access: LIB NBS*NAG | See also: F01AGE
- F01AHF** Derives eigenvectors of a real symmetric matrix from eigenvectors of the tridiagonal form where the tridiagonal matrix was produced by F01AGF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AHE. | Class(es): D4c4 | Usage: CALL F01AHF (N, M1, M2, A, IA, E, Z, IZ) | On-line doc: CALL GAMSDOC F01AHF (or @PRT NAG*DOC.F01AHF) | Access:

LIB NBS*NAG | See also: F01AGF

- F01AJE** Gives the Householder reduction of a real symmetric matrix A to tridiagonal form for use in F02AME. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AJF. | Class(es): D4c1b1 | Usage: CALL F01AJE (N, TOL, A, IA, D, E, Z, IZ) | On-line doc: CALL GAMSDOC F01AJE (or @PRT NAG*DOC.F01AJE) | Access: LIB NBS*NAG | See also: F02AME
- F01AJF** Gives the Householder reduction of a real symmetric matrix A to tridiagonal form for use in F02AMF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AJE. | Class(es): D4c1b1 | Usage: CALL F01AJF (N, TOL, A, IA, D, E, Z, IZ) | On-line doc: CALL GAMSDOC F01AJF (or @PRT NAG*DOC.F01AJF) | Access: LIB NBS*NAG | See also: F02AMF
- F01AKE** Reduces a real unsymmetric matrix to upper Hessenberg form. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AKF. | Class(es): D4c1b2 | Usage: CALL F01AKE (N, K, L, A, IA, INTGER) | On-line doc: CALL GAMSDOC F01AKE (or @PRT NAG*DOC.F01AKE) | Access: LIB NBS*NAG | See also: F01APE
- F01AKF** Reduces a real unsymmetric matrix to upper Hessenberg form. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AKE. | Class(es): D4c1b2 | Usage: CALL F01AKF (N, K, L, A, IA, INTGER) | On-line doc: CALL GAMSDOC F01AKF (or @PRT NAG*DOC.F01AKF) | Access: LIB NBS*NAG | See also: F01APF
- F01ALE** Transforms eigenvectors of a Hessenberg matrix to those of a real unsymmetric matrix from which the Hessenberg matrix has previously been derived. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01ALF. | Class(es): D4c4 | Usage: CALL F01ALE (K, L, IR, A, IA, INTGER, Z, IZ, N) | On-line doc: CALL GAMSDOC F01ALE (or @PRT NAG*DOC.F01ALE) | Access: LIB NBS*NAG
- F01ALF** Transforms eigenvectors of a Hessenberg matrix to those of a real unsymmetric matrix from which the Hessenberg matrix has previously been derived. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01ALE. | Class(es): D4c4 | Usage: CALL F01ALF (K, L, IR, A, IA, INTGER, Z, IZ, N) | On-line doc: CALL GAMSDOC F01ALF (or @PRT NAG*DOC.F01ALF) | Access: LIB NBS*NAG
- F01AME** Reduces a complex unsymmetric matrix to complex upper Hessenberg form. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AMF. | Class(es): D4c1b2 | Usage: CALL F01AME (N, K, L, AR, IAR, AI, IAI, INTGER) | On-line doc: CALL GAMSDOC F01AME (or @PRT NAG*DOC.F01AME) | Access: LIB NBS*NAG
- F01AMF** Reduces a complex unsymmetric matrix to complex upper Hessenberg form. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AME. | Class(es): D4c1b2 | Usage: CALL F01AMF (N, K, L, AR, IAR, AI, IAI, INTGER) | On-line doc: CALL GAMSDOC F01AMF (or @PRT NAG*DOC.F01AMF) | Access: LIB NBS*NAG
- F01ANE** Transforms eigenvectors of a complex upper Hessenberg matrix to those of a complex unsymmetric matrix from which the Hessenberg matrix has previously been derived. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01ANF. | Class(es): D4c4 | Usage: CALL F01ANE (K, L, IR, AR, IAR, AI, IAI, INTGER, ZR, IZR, ZI, IZI, N) | On-line doc: CALL GAMSDOC F01ANE (or @PRT NAG*DOC.F01ANE) | Access: LIB NBS*NAG
- F01ANF** Transforms eigenvectors of a complex upper Hessenberg matrix to those of a complex unsymmetric matrix from which the Hessenberg matrix has previously been derived. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01ANE. | Class(es): D4c4 | Usage: CALL F01ANF (K, L, IR, AR, IAR, AI, IAI, INTGER, ZR, IZR, ZI, IZI, N) | On-line doc: CALL GAMSDOC F01ANF (or @PRT NAG*DOC.F01ANF) | Access: LIB NBS*NAG
- F01APE** Forms the matrix of accumulated transformations from information left by F01AKE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01APF. | Class(es): D4c4 | Usage: CALL F01APE (N, K, L, INTGER, H, IH, V, IV) | On-line doc: CALL GAMSDOC F01APE (or @PRT NAG*DOC.F01APE) | Access: LIB NBS*NAG | See also: F01AKE
- F01APF** Forms the matrix of accumulated transformations from information left by F01AKF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01APE. | Class(es): D4c4 | Usage: CALL F01APF (N, K, L, INTGER, H, IH, V, IV) | On-line doc: CALL GAMSDOC F01APF (or @PRT NAG*DOC.F01APF) | Access: LIB NBS*NAG
- F01ATE** Balances a real unsymmetric matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01ATF. | Class(es): D4c1a | Usage: CALL F01ATE (N, IB, A, IA, K, L, D) | On-line doc: CALL GAMSDOC F01ATE (or @PRT NAG*DOC.F01ATE) | Access: LIB NBS*NAG
- F01ATF** Balances a real unsymmetric matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01ATE. | Class(es): D4c1a | Usage: CALL F01ATF (N, IB, A, IA, K, L, D) | On-line doc: CALL GAMSDOC F01ATF (or @PRT NAG*DOC.F01ATF) | Access: LIB NBS*NAG
- F01AUE** Transforms eigenvectors of a balanced matrix to those of the original real unsymmetric matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AUF. | Class(es): D4c4 | Usage: CALL F01AUE (N, K, L, M, D, Z, IZ) | On-line doc: CALL GAMSDOC F01AUE (or @PRT NAG*DOC.F01AUE) | Access: LIB NBS*NAG
- F01AUF** Transforms eigenvectors of a balanced matrix to those of the original real unsymmetric matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AUE. | Class(es): D4c4 | Usage: CALL F01AUF (N, K, L, M, D, Z, IZ) | On-line doc: CALL GAMSDOC F01AUF (or @PRT NAG*DOC.F01AUF) | Access: LIB NBS*NAG
- F01AVE** Balances a complex matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AVF. | Class(es): D4c1a | Usage: CALL F01AVE (N, IB, AR, IAR, AI, IAI, K, L, D) | On-line doc: CALL GAMSDOC F01AVE (or @PRT NAG*DOC.F01AVE) | Access: LIB NBS*NAG
- F01AVF** Balances a complex matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AVE.

| Class(es): D4c1a | Usage: CALL F01AVF (N, IB, AR, IAR, AI, IAI, K, L, D) | On-line doc: CALL GAMSDOC F01AVF (or @PRT NAG*DOC.F01AVF) | Access: LIB NBS*NAG

F01AWE Transforms eigenvectors of a balanced matrix to those of the original complex matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AWF. | Class(es): D4c4 | Usage: CALL F01AWE (N, K, L, M, D, ZR, IZR, ZI, IZI) | On-line doc: CALL GAMSDOC F01AWE (or @PRT NAG*DOC.F01AWE) | Access: LIB NBS*NAG

F01AWF Transforms eigenvectors of a balanced matrix to those of the original complex matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AWE. | Class(es): D4c4 | Usage: CALL F01AWF (N, K, L, M, D, ZR, IZR, ZI, IZI) | On-line doc: CALL GAMSDOC F01AWF (or @PRT NAG*DOC.F01AWF) | Access: LIB NBS*NAG

F01AXE Reduces an $m \times n$ real matrix, $m.GE.n$, to upper triangular form for use in F04AME and F04ANE. Uses Householder transformations with column pivoting. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AXF. | Class(es): D5 | Usage: CALL F01AXE (M, N, QR, IQR, ALPHA, IPIV, Y, E, IFAIL) | On-line doc: CALL GAMSDOC F01AXE (or @PRT NAG*DOC.F01AXE) | Access: LIB NBS*NAG | See also: F04AME, F04ANE

F01AXF Reduces an $m \times n$ real matrix, $m.GE.n$, to upper triangular form for use in F04AMF and F04ANF. Uses Householder transformations with column pivoting. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AXE. | Class(es): D5 | Usage: CALL F01AXF (M, N, QR, IQR, ALPHA, IPIV, Y, E, IFAIL) | On-line doc: CALL GAMSDOC F01AXF (or @PRT NAG*DOC.F01AXF) | Access: LIB NBS*NAG | See also: F04AMF, F04ANF

F01AYE Gives the Householder reduction of a real symmetric matrix A to tridiagonal form for use in F02BEE, F02AVE, and F02BFE. More economical in storage than similar F01AGE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AYF. | Class(es): D4c1b1 | Usage: CALL F01AYE (N, TOL, A, IA, D, E, E2) | On-line doc: CALL GAMSDOC F01AYE (or @PRT NAG*DOC.F01AYE) | Access: LIB NBS*NAG | See also: F02BEE, F02AVE, F02BFE

F01AYF Gives the Householder reduction of a real symmetric matrix A to tridiagonal form for use in F02BEF, F02AVF, and F02BFF. More economical in storage than similar F01AGF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AYE. | Class(es): D4c1b1 | Usage: CALL F01AYF (N, TOL, A, IA, D, E, E2) | On-line doc: CALL GAMSDOC F01AYF (or @PRT NAG*DOC.F01AYF) | Access: LIB NBS*NAG | See also: F02BEF, F02AVF, F02BFF

F01AZE Derives eigenvectors of a real symmetric matrix from eigenvectors of the tridiagonal form produced by F01AYE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AZF. | Class(es): D4c4 | Usage: CALL F01AZE (N, M1, M2, A, IA, Z, IZ) | On-line doc: CALL GAMSDOC F01AZE (or @PRT NAG*DOC.F01AZE) | Access: LIB NBS*NAG | See also: F01AYE

F01AZF Derives eigenvectors of a real symmetric matrix from eigenvectors of the tridiagonal form produced by F01AYF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AZE. | Class(es): D4c4 | Usage: CALL F01AZF (N, M1, M2, A, IA, Z, IZ) | On-line doc: CALL GAMSDOC F01AZF (or @PRT NAG*DOC.F01AZF) | Access: LIB NBS*NAG | See also: F01AYF

F01BCE Gives the Householder reduction of a complex Hermitian matrix to tridiagonal form for use in F01AVE or F02AYE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BCF. | Class(es): D4c1b1 | Usage: CALL F01BCE (N, TOL, A, IA, B, IB, D, E, WK1, WK2) | On-line doc: CALL GAMSDOC F01BCE (or @PRT NAG*DOC.F01BCE) | Access: LIB NBS*NAG | See also: F01AVE, F02AYE

F01BCF Gives the Householder reduction of a complex Hermitian matrix to tridiagonal form for use in F01AVF or F02AYF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BCE. | Class(es): D4c1b1 | Usage: CALL F01BCF (N, TOL, A, IA, B, IB, D, E, WK1, WK2) | On-line doc: CALL GAMSDOC F01BCF (or @PRT NAG*DOC.F01BCF) | Access: LIB NBS*NAG | See also: F01AVF, F02AYF

F01BDE Reduces eigenproblems $A \cdot B \cdot x = \lambda \cdot x$, $x(\text{transpose}) \cdot B \cdot A = \lambda \cdot x(\text{transpose})$, $B \cdot A \cdot y = \lambda \cdot y$, $y(\text{transpose}) \cdot A \cdot B = \lambda \cdot y(\text{transpose})$ to standard symmetric eigenproblem $Q \cdot z = \lambda \cdot z$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BDF. | Class(es): D4c1c | Usage: CALL F01BDE (N, A, IA, B, IB, DL, IFAIL) | On-line doc: CALL GAMSDOC F01BDE (or @PRT NAG*DOC.F01BDE) | Access: LIB NBS*NAG

F01BDF Reduces the eigenproblems $A \cdot B \cdot x = \lambda \cdot x$, $x(\text{transpose}) \cdot B \cdot A = \lambda \cdot x(\text{transpose})$, $B \cdot A \cdot y = \lambda \cdot y$, $y(\text{transpose}) \cdot A \cdot B = \lambda \cdot y(\text{transpose})$ to standard symmetric eigenproblem $Q \cdot z = \lambda \cdot z$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BDE. | Class(es): D4c1c | Usage: CALL F01BDF (N, A, IA, B, IB, DL, IFAIL) | On-line doc: CALL GAMSDOC F01BDF (or @PRT NAG*DOC.F01BDF) | Access: LIB NBS*NAG

F01BEE Derives eigenvectors y of the problems $y(\text{transpose}) \cdot A \cdot B = \lambda \cdot y(\text{transpose})$ and $B \cdot A \cdot y = \lambda \cdot y$ from corresponding eigenvectors of derived standard symmetric eigenproblem. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BEF. | Class(es): D4c4 | Usage: CALL F01BEE (N, M1, M2, B, IB, DL, V, IV) | On-line doc: CALL GAMSDOC F01BEE (or @PRT NAG*DOC.F01BEE) | Access: LIB NBS*NAG

F01BEF Derives eigenvectors y of the problems $y(\text{transpose}) \cdot A \cdot B = \lambda \cdot y(\text{transpose})$ and $B \cdot A \cdot y = \lambda \cdot y$ from the corresponding eigenvectors of the derived standard symmetric eigenproblem. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BEE. | Class(es): D4c4 | Usage: CALL F01BEF (N, M1, M2, B, IB, DL, V, IV) | On-line doc: CALL GAMSDOC F01BEF (or @PRT NAG*DOC.F01BEF) | Access: LIB NBS*NAG

F01BLE Calculates the rank and pseudo-inverse of an $m \times n$ real matrix A, $m.GE.n$, $\text{rank}(A).LE.n$, using Householder's factorisation. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BLF. | Class(es): D9 | Usage: CALL F01BLE (M, N, T, A, IA, AIJMAX, IRANK, INC, D, U, IU, DU, IFAIL) | On-line doc: CALL GAMSDOC F01BLE (or @PRT NAG*DOC.F01BLE) | Access: LIB NBS*NAG

F01BLF Calculates the rank and pseudo-inverse of an $m \times n$ real matrix A, $m.GE.n$, $\text{rank}(A).LE.n$, using Householder's factorisation. |

- Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BLE. | Class(es): D0 | Usage: CALL F01BLF (M, N, T, A, IA, AIJMAX, IRANK, INC, D, U, IU, DU, IFAIL) | On-line doc: CALL GAMSDOC F01BLF (or @PRT NAG*DOC.F01BLF) | Access: LIB NBS*NAG
- F01BNE** Performs the Cholesky decomposition of a complex positive definite Hermitian matrix, given the lower triangle of the matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BNF. | Class(es): D2d1b | Usage: CALL F01BNE (N, A, IA, P, IFAIL) | On-line doc: CALL GAMSDOC F01BNE (or @PRT NAG*DOC.F01BNE) | Access: LIB NBS*NAG
- F01BNF** Performs the Cholesky decomposition of a complex positive definite Hermitian matrix, given the lower triangle of the matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BNE. | Class(es): D2d1b | Usage: CALL F01BNF (N, A, IA, P, IFAIL) | On-line doc: CALL GAMSDOC F01BNF (or @PRT NAG*DOC.F01BNF) | Access: LIB NBS*NAG
- F01BPE** Determines the inverse of a complex positive definite Hermitian matrix, given the lower triangle of the matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BPF. | Class(es): D2d1b | Usage: CALL F01BPE (N, A, IA, V, IFAIL) | On-line doc: CALL GAMSDOC F01BPE (or @PRT NAG*DOC.F01BPE) | Access: LIB NBS*NAG
- F01BPF** Determines the inverse of a complex positive definite Hermitian matrix, given the lower triangle of the matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BPE. | Class(es): D2d1b | Usage: CALL F01BPF (N, A, IA, V, IFAIL) | On-line doc: CALL GAMSDOC F01BPF (or @PRT NAG*DOC.F01BPF) | Access: LIB NBS*NAG
- F01BQE** Forms the Cholesky decomposition of a real symmetric matrix G whose lower triangle only is stored. If G not positive definite, forms Cholesky decomposition of G+E, E a diagonal matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BQF. | Class(es): D2b1b | Usage: CALL F01BQE (N, EPS, RL, IRL, D, IFAIL) | On-line doc: CALL GAMSDOC F01BQE (or @PRT NAG*DOC.F01BQE) | Access: LIB NBS*NAG
- F01BQF** Forms the Cholesky decomposition of a real symmetric matrix G whose lower triangle only is stored. If G not positive definite, forms the Cholesky decomposition of G+E, E a diagonal matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BQE. | Class(es): D2b1b | Usage: CALL F01BQF (N, EPS, RL, IRL, D, IFAIL) | On-line doc: CALL GAMSDOC F01BQF (or @PRT NAG*DOC.F01BQF) | Access: LIB NBS*NAG
- F01BRE** Decomposes real sparse matrix. Either forms LU-decomposition of permutation of entire matrix, or permutes matrix to block lower triangular form and then decomposes diagonal blocks. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BRF. | Class(es): D2a4 | Usage: CALL F01BRE (N, NZ, A, LICN, IRN, LIRN, ICN, PIVOT, IKEEP, IW, W, LBLOCK, GROW, ABORT, IDISP, IFAIL) | On-line doc: CALL GAMSDOC F01BRE (or @PRT NAG*DOC.F01BRE) | Access: LIB NBS*NAG
- F01BRF** Decomposes a real sparse matrix. Either forms LU-decomposition of permutation of entire matrix, or permutes matrix to block lower triangular form, and then decomposes the diagonal blocks. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BRE. | Class(es): D2a4 | Usage: CALL F01BRF (N, NZ, A, LICN, IRN, LIRN, ICN, PIVOT, IKEEP, IW, W, LBLOCK, GROW, ABORT, IDISP, IFAIL) | On-line doc: CALL GAMSDOC F01BRF (or @PRT NAG*DOC.F01BRF) | Access: LIB NBS*NAG
- F01BSE** Decomposes a real sparse matrix using the pivotal sequence previously obtained by F01BRE when a matrix of the same sparsity pattern was decomposed. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BSF. | Class(es): D2a4 | Usage: CALL F01BSE (N, NZ, A, LICN, IVECT, JVECT, ICN, IKEEP, IW, W, GROW, ETA, RPMIN, ABORT, IDISP, IFAIL) | On-line doc: CALL GAMSDOC F01BSE (or @PRT NAG*DOC.F01BSE) | Access: LIB NBS*NAG | See also: F01BRE
- F01BSF** Decomposes a real sparse matrix using the pivotal sequence previously obtained by F01BRF when a matrix of the same sparsity pattern was decomposed. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BSE. | Class(es): D2a4 | Usage: CALL F01BSF (N, NZ, A, LICN, IVECT, JVECT, ICN, IKEEP, IW, W, GROW, ETA, RPMIN, ABORT, IDISP, IFAIL) | On-line doc: CALL GAMSDOC F01BSF (or @PRT NAG*DOC.F01BSF) | Access: LIB NBS*NAG | See also: F01BRF
- F01BTE** Decomposes a real matrix into product of triangular matrices LU by Gaussian elimination with partial pivoting. Block-column method used for efficiency on paged virtual machines. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BTF. | Class(es): D2a1 | Usage: CALL F01BTE (N, A, IA, P, DP, IFAIL) | On-line doc: CALL GAMSDOC F01BTE (or @PRT NAG*DOC.F01BTE) | Access: LIB NBS*NAG
- F01BTF** Decomposes a real matrix into a product of triangular matrices LU by Gaussian elimination with partial pivoting. Block-column method for efficiency on paged virtual machines. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BTE. | Class(es): D2a1 | Usage: CALL F01BTF (N, A, IA, P, DP, IFAIL) | On-line doc: CALL GAMSDOC F01BTF (or @PRT NAG*DOC.F01BTF) | Access: LIB NBS*NAG
- F01BUE** Decomposes a symmetric positive definite matrix into form $U*L*D*L^t*U^t$ where U is unit upper triangular, L is unit lower triangular, D is diagonal. Precedes F01BVE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BUF. | Class(es): D2b1b | Usage: CALL F01BUE (N, M1, K, A, IA, W, IFAIL) | On-line doc: CALL GAMSDOC F01BUE (or @PRT NAG*DOC.F01BUE) | Access: LIB NBS*NAG
- F01BUF** Decomposes a symmetric positive definite matrix into form $U*L*D*L^t*U^t$ where U is unit upper triangular, L is unit lower triangular, D is diagonal. Precedes F01BVF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BUE. | Class(es): D2b1b | Usage: CALL F01BUF (N, M1, K, A, IA, W, IFAIL) | On-line doc: CALL GAMSDOC F01BUF (or @PRT NAG*DOC.F01BUF) | Access: LIB NBS*NAG
- F01BVE** Transforms the generalized symmetric eigenproblem $A*x=\lambda*B*x$ to equivalent standard eigenproblem $C*y=\lambda*B*y$; A, B, C symmetric band matrices, B positive definite + decomposed $C*y=\lambda*B*y$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BVF. | Class(es): D4c1c | Usage: CALL F01BVE (N, MA1, MB1, M3, K, A, IA, B, IB, C, IC, W, IFAIL) | On-line doc: CALL GAMSDOC F01BVE (or @PRT NAG*DOC.F01BVE) | Access: LIB NBS*NAG | See also: F01BUE
- F01BVF** Transforms the generalized symmetric eigenproblem $A*x=\lambda*B*x$ to equivalent standard eigenproblem $C*y=\lambda*B*y$; A,

- B, C symmetric band matrices, B positive definite + decomposed $C*y = \lambda*y$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BVE. | Class(es): D4c1c | Usage: CALL F01BVF (N, MA1, MB1, M3, K, A, IA, B, IB, C, IC, W, IFAIL) | On-line doc: CALL GAMSDOC F01BVF (or @PRT NAG*DOC.F01BVF) | Access: LIB NBS*NAG | See also: F01BUF
- F01BWE** Reduces a symmetric band matrix to tridiagonal form. This routine normally used with F02AVE to find all eigenvalues of A. For selected values, use F02BME. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BWF. | Class(es): D4c1b1 | Usage: CALL F01BWE (N, M1, A, IA, D, E) | On-line doc: CALL GAMSDOC F01BWE (or @PRT NAG*DOC.F01BWE) | Access: LIB NBS*NAG | See also: F02AVE
- F01BWF** Reduces a symmetric band matrix to tridiagonal form. This routine normally used with F02AVF to find all eigenvalues of A. For selected eigenvalues, use F02BMF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BWE. | Class(es): D4c1b1 | Usage: CALL F01BWF (N, M1, A, IA, D, E) | On-line doc: CALL GAMSDOC F01BWF (or @PRT NAG*DOC.F01BWF) | Access: LIB NBS*NAG | See also: F02AVF
- F01BXE** Performs the Cholesky factorization $U(\text{transpose})*U$ of a real symmetric positive definite matrix A. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BXF. | Class(es): D2b1b | Usage: CALL F01BXE (N, A, IA, P, IFAIL) | On-line doc: CALL GAMSDOC F01BXE (or @PRT NAG*DOC.F01BXE) | Access: LIB NBS*NAG
- F01BXF** Performs the Cholesky factorization $U(\text{transpose})*U$ of a real symmetric positive definite matrix A. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BXE. | Class(es): D2b1b | Usage: CALL F01BXF (N, A, IA, P, IFAIL) | On-line doc: CALL GAMSDOC F01BXF (or @PRT NAG*DOC.F01BXF) | Access: LIB NBS*NAG
- F01CAE** Sets elements of an $m*n$ matrix A to zero. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CAF. | Class(es): D1b1 | Usage: CALL F01CAE (A, M, N, IFAIL) | On-line doc: CALL GAMSDOC F01CAE (or @PRT NAG*DOC.F01CAE) | Access: LIB NBS*NAG
- F01CAF** Sets elements of an $m*n$ matrix A to zero. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CAE. | Class(es): D1b1 | Usage: CALL F01CAF (A, M, N, IFAIL) | On-line doc: CALL GAMSDOC F01CAF (or @PRT NAG*DOC.F01CAF) | Access: LIB NBS*NAG
- F01CBE** Sets the elements $A(i,j)$ to one if $i=j$ and zero otherwise, where $1 \leq i \leq m$ and $1 \leq j \leq n$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CBF. | Class(es): D1b1 | Usage: CALL F01CBE (A, M, N, IFAIL) | On-line doc: CALL GAMSDOC F01CBE (or @PRT NAG*DOC.F01CBE) | Access: LIB NBS*NAG
- F01CBF** Sets the elements $A(i,j)$ to one if $i=j$ and zero otherwise, where $1 \leq i \leq m$ and $1 \leq j \leq n$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CBE. | Class(es): D1b1 | Usage: CALL F01CBF (A, M, N, IFAIL) | On-line doc: CALL GAMSDOC F01CBF (or @PRT NAG*DOC.F01CBF) | Access: LIB NBS*NAG
- F01CDE** Adds elements of the $m*n$ matrices B and C and stores the results in elements of the matrix A. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CDF. | Class(es): D1b5 | Usage: CALL F01CDE (A, B, C, M, N, IFAIL) | On-line doc: CALL GAMSDOC F01CDE (or @PRT NAG*DOC.F01CDE) | Access: LIB NBS*NAG
- F01CDF** Adds elements of the $m*n$ matrices B and C and stores the results in elements of the matrix A. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CDE. | Class(es): D1b5 | Usage: CALL F01CDF (A, B, C, M, N, IFAIL) | On-line doc: CALL GAMSDOC F01CDF (or @PRT NAG*DOC.F01CDF) | Access: LIB NBS*NAG
- F01CEE** Subtracts elements of the matrix C from elements of the matrix B and stores the results in elements of the matrix A. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CEF. | Class(es): D1b5 | Usage: CALL F01CEE (A, B, C, M, N, IFAIL) | On-line doc: CALL GAMSDOC F01CEE (or @PRT NAG*DOC.F01CEE) | Access: LIB NBS*NAG
- F01CEF** Subtracts elements of the matrix C from elements of the matrix B and stores the results in elements of the matrix A. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CEE. | Class(es): D1b5 | Usage: CALL F01CEF (A, B, C, M, N, IFAIL) | On-line doc: CALL GAMSDOC F01CEF (or @PRT NAG*DOC.F01CEF) | Access: LIB NBS*NAG
- F01CFE** Copies elements of the matrix B into different positions in the matrix A. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CFF. | Class(es): D1b8 | Usage: CALL F01CFE (A, MA, NA, P, Q, B, MB, NB, M1, M2, N1, N2, IFAIL) | On-line doc: CALL GAMSDOC F01CFE (or @PRT NAG*DOC.F01CFE) | Access: LIB NBS*NAG
- F01CFF** Copies elements of the matrix B into different positions in the matrix A. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CFE. | Class(es): D1b8 | Usage: CALL F01CFF (A, MA, NA, P, Q, B, MB, NB, M1, M2, N1, N2, IFAIL) | On-line doc: CALL GAMSDOC F01CFF (or @PRT NAG*DOC.F01CFF) | Access: LIB NBS*NAG
- F01CGE** Adds elements of the matrix B to elements in different positions in the matrix A. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CGF. | Class(es): D1b5 | Usage: CALL F01CGE (A, MA, NA, P, Q, B, MB, NB, M1, M2, N1, N2, IFAIL) | On-line doc: CALL GAMSDOC F01CGE (or @PRT NAG*DOC.F01CGE) | Access: LIB NBS*NAG
- F01CGF** Adds elements of the matrix B to elements in different positions in the matrix A. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CGE. | Class(es): D1b5 | Usage: CALL F01CGF (A, MA, NA, P, Q, B, MB, NB, M1, M2, N1, N2, IFAIL) | On-line doc: CALL GAMSDOC F01CGF (or @PRT NAG*DOC.F01CGF) | Access: LIB NBS*NAG
- F01CHE** Subtracts elements of the matrix B from elements in a different position in the matrix A. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CHF. | Class(es): D1b5 | Usage: CALL F01CHE (A, MA, NA, P, Q, B, MB, NB, M1, M2, N1, N2, IFAIL) | On-line doc: CALL GAMSDOC F01CHE (or @PRT NAG*DOC.F01CHE) | Access: LIB NBS*NAG

- F01CHF** Subtracts elements of the matrix B from elements in a different position in the matrix A. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CHE. | Class(es): D1b5 | Usage: CALL F01CHF (A, MA, NA, P, Q, B, MB, NB, M1, M2, N1, N2, IFAIL) | On-line doc: CALL GAMSDOC F01CHF (or @PRT NAG*DOC.F01CHF) | Access: LIB NBS*NAG
- F01CKE** Returns with the result of multiplication of two matrices B and C in the matrix A, with the option to overwrite B or C. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CKF. | Class(es): D1b6 | Usage: CALL F01CKE (A, B, C, N, P, M, Z, IZ, OPT, IFAIL) | On-line doc: CALL GAMSDOC F01CKE (or @PRT NAG*DOC.F01CKE) | Access: LIB NBS*NAG
- F01CKF** Returns with the result of multiplication of two matrices B and C in the matrix A, with the option to overwrite B or C. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CKE. | Class(es): D1b6 | Usage: CALL F01CKF (A, B, C, N, P, M, Z, IZ, OPT, IFAIL) | On-line doc: CALL GAMSDOC F01CKF (or @PRT NAG*DOC.F01CKF) | Access: LIB NBS*NAG
- F01CLE** Post-multiplies the matrix B with the transpose of the matrix C and places the result in the matrix A. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CLF. | Class(es): D1b6 | Usage: CALL F01CLE (A, B, C, N, P, M, IFAIL) | On-line doc: CALL GAMSDOC F01CLE (or @PRT NAG*DOC.F01CLE) | Access: LIB NBS*NAG
- F01CLF** Post-multiplies the matrix B with the transpose of the matrix C and places the result in the matrix A. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CLE. | Class(es): D1b6 | Usage: CALL F01CLF (A, B, C, N, P, M, IFAIL) | On-line doc: CALL GAMSDOC F01CLF (or @PRT NAG*DOC.F01CLF) | Access: LIB NBS*NAG
- F01CME** Copies elements of one matrix into a second matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CMF. | Class(es): D1b8 | Usage: CALL F01CME (A, LA, B, LB, M, N) | On-line doc: CALL GAMSDOC F01CME (or @PRT NAG*DOC.F01CME) | Access: LIB NBS*NAG
- F01CMF** Copies elements of one matrix into a second matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CME. | Class(es): D1b8 | Usage: CALL F01CMF (A, LA, B, LB, M, N) | On-line doc: CALL GAMSDOC F01CMF (or @PRT NAG*DOC.F01CMF) | Access: LIB NBS*NAG
- F01CNE** Copies a vector of length M into a row of a matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CNF. | Class(es): D1a5 D1b8 | Usage: CALL F01CNE (V, M, A, LA, I) | On-line doc: CALL GAMSDOC F01CNE (or @PRT NAG*DOC.F01CNE) | Access: LIB NBS*NAG
- F01CNF** Copies a vector of length M into a row of a matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CNE. | Class(es): D1a5 D1b8 | Usage: CALL F01CNF (V, M, A, LA, I) | On-line doc: CALL GAMSDOC F01CNF (or @PRT NAG*DOC.F01CNF) | Access: LIB NBS*NAG
- F01CPE** Copies the contents of a vector into a second vector. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CPF. | Class(es): D1a5 | Usage: CALL F01CPE (A, B, N) | On-line doc: CALL GAMSDOC F01CPE (or @PRT NAG*DOC.F01CPE) | Access: LIB NBS*NAG
- F01CPF** Copies the contents of a vector into a second vector. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CPE. | Class(es): D1a5 | Usage: CALL F01CPF (A, B, N) | On-line doc: CALL GAMSDOC F01CPF (or @PRT NAG*DOC.F01CPF) | Access: LIB NBS*NAG
- F01CQE** Sets the elements of a vector to zero. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CQF. | Class(es): D1a1 | Usage: CALL F01CQE (A, N) | On-line doc: CALL GAMSDOC F01CQE (or @PRT NAG*DOC.F01CQE) | Access: LIB NBS*NAG
- F01CQF** Sets the elements of a vector to zero. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CQE. | Class(es): D1a1 | Usage: CALL F01CQF (A, N) | On-line doc: CALL GAMSDOC F01CQF (or @PRT NAG*DOC.F01CQF) | Access: LIB NBS*NAG
- F01CRE** Re-orders the elements of a vector of length m*n, containing an m*n matrix, A, so that the new vector contains the transpose matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CRF. | Class(es): D1b3 | Usage: CALL F01CRE(A, M, N, MN, MOVE, LMOVE, IFAIL) | On-line doc: CALL GAMSDOC F01CRE (or @PRT NAG*DOC.F01CRE) | Access: LIB NBS*NAG
- F01CRF** Re-orders the elements of a vector of length m*n, containing an m*n matrix, A, so that the new vector contains the transpose matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CRE. | Class(es): D1b3 | Usage: CALL F01CRF(A, M, N, MN, MOVE, LMOVE, IFAIL) | On-line doc: CALL GAMSDOC F01CRF (or @PRT NAG*DOC.F01CRF) | Access: LIB NBS*NAG
- F01CSE** Forms the product $c=A*b$ where b is a vector and A is a symmetric matrix whose lower triangle is stored by rows in a one-dimensional array. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CSF. | Class(es): D1b4 | Usage: CALL F01CSE (A, LA, B, N, C) | On-line doc: CALL GAMSDOC F01CSE (or @PRT NAG*DOC.F01CSE) | Access: LIB NBS*NAG
- F01CSF** Forms the product $c = A*b$ where b is a vector and A is a symmetric matrix whose lower triangle is stored by rows in a one-dimensional array. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CSE. | Class(es): D1b4 | Usage: CALL F01CSF (A, LA, B, N, C) | On-line doc: CALL GAMSDOC F01CSF (or @PRT NAG*DOC.F01CSF) | Access: LIB NBS*NAG
- F01DAE** Returns the sum of an initial value and a scalar product, using basic precision arithmetic. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01DAF. | Class(es): D1a4 | Usage: D = F01DAE (L, M, C1, IRA, ICB, A, IA, B, IB, N) | On-line doc: CALL GAMSDOC F01DAE (or @PRT NAG*DOC.F01DAE) | Access: LIB NBS*NAG
- F01DAF** Returns the sum of an initial value and a scalar product, using basic precision arithmetic. | Proprietary double precision Fortran

- subprogram in NAG library. Single precision version is F01DAE. | Class(es): D1a4 | Usage: D = F01DAF (L, M, C1, IRA, ICB, A, IA, B, IB, N) | On-line doc: CALL GAMSDOC F01DAF (or @PRT NAG*DOC.F01DAF) | Access: LIB NBS*NAG
- F01DBE** Returns the sum of an initial value and a scalar product, using additional precision arithmetic. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01DBF. | Class(es): D1a4 | Usage: D = F01DBE (L, M, C1, IRA, ICB, A, IA, B, IB, N) | On-line doc: CALL GAMSDOC F01DBE (or @PRT NAG*DOC.F01DBE) | Access: LIB NBS*NAG
- F01DBF** Returns the sum of an initial value and a scalar product, using additional precision arithmetic. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01DBE. | Class(es): D1a4 | Usage: D = F01DBF (L, M, C1, IRA, ICB, A, IA, B, IB, N) | On-line doc: CALL GAMSDOC F01DBF (or @PRT NAG*DOC.F01DBF) | Access: LIB NBS*NAG
- F01DCE** Computes the value of a complex scalar product and subtracts it from a complex initial value, using basic precision arithmetic. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01DCF. | Class(es): D1a4 | Usage: CALL F01DCE (L, M, CX, IRA, ICB, A, IA, B, IB, N, CR, C1) | On-line doc: CALL GAMSDOC F01DCE (or @PRT NAG*DOC.F01DCE) | Access: LIB NBS*NAG
- F01DCF** Computes the value of a complex scalar product and subtracts it from a complex initial value, using basic precision arithmetic. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01DCE. | Class(es): D1a4 | Usage: CALL F01DCF (L, M, CX, IRA, ICB, A, IA, B, IB, N, CR, C1) | On-line doc: CALL GAMSDOC F01DCF (or @PRT NAG*DOC.F01DCF) | Access: LIB NBS*NAG
- F01DDE** Computes the value of a complex scalar product and subtracts it from a complex initial value, using additional precision arithmetic. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01DDF. | Class(es): D1a4 | Usage: CALL F01DDE (L, M, CX, IRA, ICB, A, IA, B, IB, N, CR, C1) | On-line doc: CALL GAMSDOC F01DDE (or @PRT NAG*DOC.F01DDE) | Access: LIB NBS*NAG
- F01DDF** Computes the value of a complex scalar product and subtracts it from a complex initial value, using additional precision arithmetic. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01DDE. | Class(es): D1a4 | Usage: CALL F01DDF (L, M, CX, IRA, ICB, A, IA, B, IB, N, CR, C1) | On-line doc: CALL GAMSDOC F01DDF (or @PRT NAG*DOC.F01DDF) | Access: LIB NBS*NAG
- F01DEE** Returns the value of the scalar product of two arrays of length N, using basic precision arithmetic. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01DEF. | Class(es): D1a4 | Usage: D = F01DEE (A, B, N) | On-line doc: CALL GAMSDOC F01DEE (or @PRT NAG*DOC.F01DEE) | Access: LIB NBS*NAG
- F01DEF** Returns the value of the scalar product of two arrays of length N, using basic precision arithmetic. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01DEE. | Class(es): D1a4 | Usage: D = F01DEF (A, B, N) | On-line doc: CALL GAMSDOC F01DEF (or @PRT NAG*DOC.F01DEF) | Access: LIB NBS*NAG
- F01LBE** LU-factorisation, real band matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01LBF. | Class(es): D2a2 | Usage: CALL F01LBE (N, M1, M2, A, IA, AL, IL, IN, IV, IFAIL) | On-line doc: CALL GAMSDOC F01LBE (or @PRT NAG*DOC.F01LBE) | Access: LIB NBS*NAG
- F01LBF** LU-factorisation, real band matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01LBE. | Class(es): D2a2 | Usage: CALL F01LBF (N, M1, M2, A, IA, AL, IL, IN, IV, IFAIL) | On-line doc: CALL GAMSDOC F01LBF (or @PRT NAG*DOC.F01LBF) | Access: LIB NBS*NAG
- F01LZE** Reduction by similarity transformations, real matrix to bidiagonal form. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01LZF. | Class(es): D6 | Usage: CALL F01LZE (N, A, NRA, C, NRC, WANTB, B, WANTQ, WANTY, Y, NRY, LY, WANTZ, Z, NRZ, NCZ, D, E, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC F01LZE (or @PRT NAG*DOC.F01LZE) | Access: LIB NBS*NAG
- F01LZF** Reduction by similarity transformations, real matrix to bidiagonal form. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01LZE. | Class(es): D6 | Usage: CALL F01LZF (N, A, NRA, C, NRC, WANTB, B, WANTQ, WANTY, Y, NRY, LY, WANTZ, Z, NRZ, NCZ, D, E, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC F01LZF (or @PRT NAG*DOC.F01LZF) | Access: LIB NBS*NAG
- F01MCE** LDLt-factorisation, real symmetric positive-definite variable-bandwidth matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01MCF. | Class(es): D2b2 | Usage: CALL F01MCE (N, A, LAL, NROW, AL, D, IFAIL) | On-line doc: CALL GAMSDOC F01MCE (or @PRT NAG*DOC.F01MCE) | Access: LIB NBS*NAG
- F01MCF** LDLt-factorisation, real symmetric positive-definite variable-bandwidth matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01MCE. | Class(es): D2b2 | Usage: CALL F01MCF (N, A, LAL, NROW, AL, D, IFAIL) | On-line doc: CALL GAMSDOC F01MCF (or @PRT NAG*DOC.F01MCF) | Access: LIB NBS*NAG
- F01QAE** QR-factorisation, real $m \times n$ matrix ($m \geq n$). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01QAF. | Class(es): D5 | Usage: CALL F01QAE (M, N, A, NRA, C, NRC, Z, IFAIL) | On-line doc: CALL GAMSDOC F01QAE (or @PRT NAG*DOC.F01QAE) | Access: LIB NBS*NAG
- F01QAF** QR-factorisation, real $m \times n$ matrix ($m \geq n$). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01QAE. | Class(es): D5 | Usage: CALL F01QAF (M, N, A, NRA, C, NRC, Z, IFAIL) | On-line doc: CALL GAMSDOC F01QAF (or @PRT NAG*DOC.F01QAF) | Access: LIB NBS*NAG
- F01QBE** RQ-factorisation, real $m \times n$ matrix ($m \leq n$). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01QBF. | Class(es): D5 | Usage: CALL F01QBE (M, N, A, NRA, C, NRC, WORK, IFAIL) | On-line doc: CALL GAMSDOC

- F01QBE** (or @PRT NAG*DOC.F01QBE) | Access: LIB NBS*NAG
- F01QBF** RQ-factorisation, real $m \times n$ matrix ($m \leq n$). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01QBE. | Class(es): D5 | Usage: CALL F01QBF (M, N, A, NRA, C, NRC, WORK, IFAIL) | On-line doc: CALL GAMSDOC F01QBF (or @PRT NAG*DOC.F01QBF) | Access: LIB NBS*NAG
- F02AAE** Real symmetric matrix, (black box), all eigenvalues. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AAF. | Class(es): D4a1 | Usage: CALL F02AAE (A, IA, N, R, E, IFAIL) | On-line doc: CALL GAMSDOC F02AAE (or @PRT NAG*DOC.F02AAE) | Access: LIB NBS*NAG
- F02AAF** Real symmetric matrix, (black box), all eigenvalues. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AAE. | Class(es): D4a1 | Usage: CALL F02AAF (A, IA, N, R, E, IFAIL) | On-line doc: CALL GAMSDOC F02AAF (or @PRT NAG*DOC.F02AAF) | Access: LIB NBS*NAG
- F02ABE** Real symmetric matrix, (black box), all eigenvalues and eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02ABF. | Class(es): D4a1 | Usage: CALL F02ABE (A, IA, N, R, V, IV, E, IFAIL) | On-line doc: CALL GAMSDOC F02ABE (or @PRT NAG*DOC.F02ABE) | Access: LIB NBS*NAG
- F02ABF** Real symmetric matrix, (black box), all eigenvalues and eigenvectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02ABE. | Class(es): D4a1 | Usage: CALL F02ABF (A, IA, N, R, V, IV, E, IFAIL) | On-line doc: CALL GAMSDOC F02ABF (or @PRT NAG*DOC.F02ABF) | Access: LIB NBS*NAG
- F02ADE** Generalised real symmetric eigenproblem $Ax=kBx$ with positive-definite B (black box), all eigenvalues. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02ADF. | Class(es): D4b1 | Usage: CALL F02ADE (A, IA, B, IB, N, R, DE, IFAIL) | On-line doc: CALL GAMSDOC F02ADE (or @PRT NAG*DOC.F02ADE) | Access: LIB NBS*NAG
- F02ADF** Generalised real symmetric eigenproblem $Ax=kBx$ with positive-definite B (black box), all eigenvalues. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02ADE. | Class(es): D4b1 | Usage: CALL F02ADF (A, IA, B, IB, N, R, DE, IFAIL) | On-line doc: CALL GAMSDOC F02ADF (or @PRT NAG*DOC.F02ADF) | Access: LIB NBS*NAG
- F02AEE** Generalised real symmetric eigenproblem $Ax=kBx$ with positive-definite B (black box), all eigenvalues and eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AEF. | Class(es): D4b1 | Usage: CALL F02AEE (A, IA, B, IB, N, R, V, IV, DL, E, IFAIL) | On-line doc: CALL GAMSDOC F02AEE (or @PRT NAG*DOC.F02AEE) | Access: LIB NBS*NAG
- F02AEF** Generalised real symmetric eigenproblem $Ax=kBx$ with positive-definite B (black box), all eigenvalues and eigenvectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AEE. | Class(es): D4b1 | Usage: CALL F02AEF (A, IA, B, IB, N, R, V, IV, DL, E, IFAIL) | On-line doc: CALL GAMSDOC F02AEF (or @PRT NAG*DOC.F02AEF) | Access: LIB NBS*NAG
- F02AFE** Real matrix, (black box), all eigenvalues. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AFF. | Class(es): D4a2 | Usage: CALL F02AFE (A, IA, N, RR, RI, INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AFE (or @PRT NAG*DOC.F02AFE) | Access: LIB NBS*NAG
- F02AFF** Real matrix, (black box), all eigenvalues. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AFE. | Class(es): D4a2 | Usage: CALL F02AFF (A, IA, N, RR, RI, INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AFF (or @PRT NAG*DOC.F02AFF) | Access: LIB NBS*NAG
- F02AGE** Real matrix, (black box), all eigenvalues and eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AGF. | Class(es): D4a2 | Usage: CALL F02AGE (A, IA, N, RR, RI, VR, IVR, VI, IVI, 1 INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AGE (or @PRT NAG*DOC.F02AGE) | Access: LIB NBS*NAG
- F02AGF** Real matrix, (black box), all eigenvalues and eigenvectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AGE. | Class(es): D4a2 | Usage: CALL F02AGF (A, IA, N, RR, RI, VR, IVR, VI, IVI, 1 INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AGF (or @PRT NAG*DOC.F02AGF) | Access: LIB NBS*NAG
- F02AJE** Complex matrix, (black box), all eigenvalues. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AJF. | Class(es): D4a4 | Usage: CALL F02AJE (AR, IAR, AI, IAI, N, RR, RI, INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AJE (or @PRT NAG*DOC.F02AJE) | Access: LIB NBS*NAG
- F02AJF** Complex matrix, (black box), all eigenvalues. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AJE. | Class(es): D4a4 | Usage: CALL F02AJF (AR, IAR, AI, IAI, N, RR, RI, INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AJF (or @PRT NAG*DOC.F02AJF) | Access: LIB NBS*NAG
- F02AKE** Complex matrix, (black box), all eigenvalues and eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AKF. | Class(es): D4a4 | Usage: CALL F02AKE (AR, IAR, AI, IAI, N, RR, RI, VR, IVR, VI, IVI, INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AKE (or @PRT NAG*DOC.F02AKE) | Access: LIB NBS*NAG
- F02AKF** Complex matrix, (black box), all eigenvalues and eigenvectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AKE. | Class(es): D4a4 | Usage: CALL F02AKF (AR, IAR, AI, IAI, N, RR, RI, VR, IVR, VI, IVI, INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AKF (or @PRT NAG*DOC.F02AKF) | Access: LIB NBS*NAG
- F02AME** Real symmetric matrix, all eigenvalues and eigenvectors, after reduction to tridiagonal form by F01AJE, QL algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AMF. | Class(es): D4c2a | Usage: CALL F02AME (N, EPS, D, E, V, IV, IFAIL) | On-line doc: CALL GAMSDOC F02AME (or @PRT NAG*DOC.F02AME) | Access: LIB NBS*NAG

See also: F01AJE

- F02AMF** Real symmetric matrix, all eigenvalues and eigenvectors, after reduction to tridiagonal form by F01AJF, QL algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AME. | Class(es): D4c2a | Usage: CALL F02AMF (N, EPS, D, E, V, IV, IFAIL) | On-line doc: CALL GAMSDOC F02AMF (or @PRT NAG*DOC.F02AMF) | Access: LIB NBS*NAG | See also: F01AJF
- F02ANE** Complex upper Hessenberg matrix, all eigenvalues, LR algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02ANF. | Class(es): D4c2b | Usage: CALL F02ANE (N, EPS, HR, IHR, HI, IHI, RR, RI, IFAIL) | On-line doc: CALL GAMSDOC F02ANE (or @PRT NAG*DOC.F02ANE) | Access: LIB NBS*NAG
- F02ANF** Complex upper Hessenberg matrix, all eigenvalues, LR algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02ANE. | Class(es): D4c2b | Usage: CALL F02ANF (N, EPS, HR, IHR, HI, IHI, RR, RI, IFAIL) | On-line doc: CALL GAMSDOC F02ANF (or @PRT NAG*DOC.F02ANF) | Access: LIB NBS*NAG
- F02APE** Real upper Hessenberg matrix, all eigenvalues, QR algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02APF. | Class(es): D4c2b | Usage: CALL F02APE (N, EPS, H, IH, RR, RI, ICNT, IFAIL) | On-line doc: CALL GAMSDOC F02APE (or @PRT NAG*DOC.F02APE) | Access: LIB NBS*NAG
- F02APF** Real upper Hessenberg matrix, all eigenvalues, QR algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02APE. | Class(es): D4c2b | Usage: CALL F02APF (N, EPS, H, IH, RR, RI, ICNT, IFAIL) | On-line doc: CALL GAMSDOC F02APF (or @PRT NAG*DOC.F02APF) | Access: LIB NBS*NAG
- F02AQE** Real matrix, all eigenvalues and eigenvectors, after reduction to upper Hessenberg form by F01AKE and F01APE, QR algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AQF. | Class(es): D4c2b | Usage: CALL F02AQE (N, K, L, EPS, H, IH, V, IV, RR, RI, INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AQE (or @PRT NAG*DOC.F02AQE) | Access: LIB NBS*NAG
- F02AQF** Real matrix, all eigenvalues and eigenvectors, after reduction to upper Hessenberg form by F01AKF and F01APF, QR algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AQE. | Class(es): D4c2b | Usage: CALL F02AQF (N, K, L, EPS, H, IH, V, IV, RR, RI, INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AQF (or @PRT NAG*DOC.F02AQF) | Access: LIB NBS*NAG
- F02ARE** Complex matrix, all eigenvalues and eigenvectors, after reduction to upper Hessenberg form by F01AME, LR algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02ARF. | Class(es): D4c2b | Usage: CALL F02ARE (N, K, L, EPS, INTGER, HR, IHR, HI, IHI, RR, RI, VR, IVR, VI, IVI, IFAIL) | On-line doc: CALL GAMSDOC F02ARE (or @PRT NAG*DOC.F02ARE) | Access: LIB NBS*NAG | See also: F01AMF
- F02ARF** Complex matrix, all eigenvalues and eigenvectors, after reduction to upper Hessenberg form by F01AMF, LR algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02ARE. | Class(es): D4c2b | Usage: CALL F02ARF (N, K, L, EPS, INTGER, HR, IHR, HI, IHI, RR, RI, VR, IVR, VI, IVI, IFAIL) | On-line doc: CALL GAMSDOC F02ARF (or @PRT NAG*DOC.F02ARF) | Access: LIB NBS*NAG | See also: F01AMF
- F02AVE** Real symmetric tridiagonal matrix, all eigenvalues, QL algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AVF. | Class(es): D4a5 | Usage: CALL F02AVE (N, EPS, D, E, IFAIL) | On-line doc: CALL GAMSDOC F02AVE (or @PRT NAG*DOC.F02AVE) | Access: LIB NBS*NAG
- F02AVF** Real symmetric tridiagonal matrix, all eigenvalues, QL algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AVE. | Class(es): D4a5 | Usage: CALL F02AVF (N, EPS, D, E, IFAIL) | On-line doc: CALL GAMSDOC F02AVF (or @PRT NAG*DOC.F02AVF) | Access: LIB NBS*NAG
- F02AWE** Complex Hermitian matrix, (black box), all eigenvalues. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AWF. | Class(es): D4a3 | Usage: CALL F02AWE (AR, IAR, AI, IAI, N, R, WK1, WK2, WK3, IFAIL) | On-line doc: CALL GAMSDOC F02AWE (or @PRT NAG*DOC.F02AWE) | Access: LIB NBS*NAG
- F02AWF** Complex Hermitian matrix, (black box), all eigenvalues. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AWE. | Class(es): D4a3 | Usage: CALL F02AWF (AR, IAR, AI, IAI, N, R, WK1, WK2, WK3, IFAIL) | On-line doc: CALL GAMSDOC F02AWF (or @PRT NAG*DOC.F02AWF) | Access: LIB NBS*NAG
- F02AXE** Complex Hermitian matrix, (black box), all eigenvalues and eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AXF. | Class(es): D4a3 | Usage: CALL F02AXE (AR, IAR, AI, IAI, N, R, VR, IVR, VI, IVI, WK1, WK2, WK3, IFAIL) | On-line doc: CALL GAMSDOC F02AXE (or @PRT NAG*DOC.F02AXE) | Access: LIB NBS*NAG
- F02AXF** Complex Hermitian matrix, (black box), all eigenvalues and eigenvectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AXE. | Class(es): D4a3 | Usage: CALL F02AXF (AR, IAR, AI, IAI, N, R, VR, IVR, VI, IVI, WK1, WK2, WK3, IFAIL) | On-line doc: CALL GAMSDOC F02AXF (or @PRT NAG*DOC.F02AXF) | Access: LIB NBS*NAG
- F02AYE** Complex Hermitian matrix, all eigenvalues and eigenvectors, after reduction to real tridiagonal form by F01BCE, QL algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AYF. | Class(es): D4c2a | Usage: CALL F02AYE (N, EPS, D, E, VR, IVR, VI, IVI, IFAIL) | On-line doc: CALL GAMSDOC F02AYE (or @PRT NAG*DOC.F02AYE) | Access: LIB NBS*NAG | See also: F01BCE
- F02AYF** Complex Hermitian matrix, all eigenvalues and eigenvectors, after reduction to real tridiagonal form by F01BCF, QL algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AYE. | Class(es): D4c2a | Usage: CALL

- F02AYF** (N, EPS, D, E, VR, IVR, VI, IVI, IFAIL) | On-line doc: CALL GAMSDOC F02AYF (or @PRT NAG*DOC.F02AYF) | Access: LIB NBS*NAG | See also: F01BCF
- F02BBE** Real symmetric matrix, (black box), selected eigenvalues and eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02BBF. | Class(es): D4a1 | Usage: CALL F02BBE (A, IA, N, ALB, UB, M, MM, R, V, IV, D, E, E2, X, G, LOG, ICOUNT, IFAIL) | On-line doc: CALL GAMSDOC F02BBE (or @PRT NAG*DOC.F02BBE) | Access: LIB NBS*NAG
- F02BBF** Real symmetric matrix, (black box), selected eigenvalues and eigenvectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02BBE. | Class(es): D4a1 | Usage: CALL F02BBF (A, IA, N, ALB, UB, M, MM, R, V, IV, D, E, E2, X, G, LOG, ICOUNT, IFAIL) | On-line doc: CALL GAMSDOC F02BBF (or @PRT NAG*DOC.F02BBF) | Access: LIB NBS*NAG
- F02BCE** Real matrix, (black box), selected eigenvalues and eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02BCF. | Class(es): D4a2 | Usage: CALL F02BCE (A, IA, N, ALB, UB, M, MM, RR, RI, VR, IVR, VI, IVI, INTGER, ICNT, C, B, IB, U, V, IFAIL) | On-line doc: CALL GAMSDOC F02BCE (or @PRT NAG*DOC.F02BCE) | Access: LIB NBS*NAG
- F02BCF** Real matrix, (black box), selected eigenvalues and eigenvectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02BCE. | Class(es): D4a2 | Usage: CALL F02BCF (A, IA, N, ALB, UB, M, MM, RR, RI, VR, IVR, VI, IVI, INTGER, ICNT, C, B, IB, U, V, IFAIL) | On-line doc: CALL GAMSDOC F02BCF (or @PRT NAG*DOC.F02BCF) | Access: LIB NBS*NAG
- F02BDE** Complex matrix, (black box), selected eigenvalues and eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02BDF. | Class(es): D4a4 | Usage: CALL F02BDE (AR, IAR, AI, IAI, N, ALB, UB, M, MM, RR, RI, VR, IVR, VI, IVI, INTGER, C, BR, IBR, BI, IBI, U, V, IFAIL) | On-line doc: CALL GAMSDOC F02BDE (or @PRT NAG*DOC.F02BDE) | Access: LIB NBS*NAG
- F02BDF** Complex matrix, (black box), selected eigenvalues and eigenvectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02BDE. | Class(es): D4a4 | Usage: CALL F02BDF (AR, IAR, AI, IAI, N, ALB, UB, M, MM, RR, RI, VR, IVR, VI, IVI, INTGER, C, BR, IBR, BI, IBI, U, V, IFAIL) | On-line doc: CALL GAMSDOC F02BDF (or @PRT NAG*DOC.F02BDF) | Access: LIB NBS*NAG
- F02BEE** Real symmetric tridiagonal matrix, selected eigenvalues and eigenvectors, bisection and inverse iteration. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02BEF. | Class(es): D4a5 | Usage: CALL F02BEE (N, D, ALB, UB, EPS, EPS1, E, E2, M, MM, R, V, IV, ICOUNT, X, LOG, IFAIL) | On-line doc: CALL GAMSDOC F02BEE (or @PRT NAG*DOC.F02BEE) | Access: LIB NBS*NAG
- F02BEF** Real symmetric tridiagonal matrix, selected eigenvalues and eigenvectors, bisection and inverse iteration. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02BEE. | Class(es): D4a5 | Usage: CALL F02BEF (N, D, ALB, UB, EPS, EPS1, E, E2, M, MM, R, V, IV, ICOUNT, X, LOG, IFAIL) | On-line doc: CALL GAMSDOC F02BEF (or @PRT NAG*DOC.F02BEF) | Access: LIB NBS*NAG
- F02BFE** Real symmetric tridiagonal matrix, selected eigenvalues, bisection. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02BFF. | Class(es): D4a5 | Usage: CALL F02BFE (D, E, E2, N, M1, M2, MM12, EPS1, EPS, EPS2, IZ, R, WU) | On-line doc: CALL GAMSDOC F02BFE (or @PRT NAG*DOC.F02BFE) | Access: LIB NBS*NAG
- F02BFF** Real symmetric tridiagonal matrix, selected eigenvalues, bisection. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02BFE. | Class(es): D4a5 | Usage: CALL F02BFF (D, E, E2, N, M1, M2, MM12, EPS1, EPS, EPS2, IZ, R, WU) | On-line doc: CALL GAMSDOC F02BFF (or @PRT NAG*DOC.F02BFF) | Access: LIB NBS*NAG
- F02BJE** Generalised eigenproblem $Ax=kBx$, QZ algorithm (black box), real matrices, all eigenvalues and (optionally) eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02BJF. | Class(es): D4b2 | Usage: CALL F02BJE (N, A, IA, B, IB, EPS1, ALFR, ALF1, BETA, MATV, V, IV, ITER, IFAIL) | On-line doc: CALL GAMSDOC F02BJE (or @PRT NAG*DOC.F02BJE) | Access: LIB NBS*NAG
- F02BJF** Generalised eigenproblem $Ax=kBx$, QZ algorithm (black box), real matrices, all eigenvalues and (optionally) eigenvectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02BJE. | Class(es): D4b2 | Usage: CALL F02BJF (N, A, IA, B, IB, EPS1, ALFR, ALF1, BETA, MATV, V, IV, ITER, IFAIL) | On-line doc: CALL GAMSDOC F02BJF (or @PRT NAG*DOC.F02BJF) | Access: LIB NBS*NAG
- F02BKE** Compute selected eigenvectors of a real upper Hessenberg matrix by inverse iteration, given estimates of their associated eigenvalues. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02BKF. | Class(es): D4c3 | Usage: CALL F02BKE (N, M, H, IH, RI, C, RR, V, IV, B, IB, U, W, IFAIL) | On-line doc: CALL GAMSDOC F02BKE (or @PRT NAG*DOC.F02BKE) | Access: LIB NBS*NAG | See also: F02AKE
- F02BKF** Compute selected eigenvectors of a real upper Hessenberg matrix by inverse iteration, given estimates of their associated eigenvalues. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02BKE. | Class(es): D4c3 | Usage: CALL F02BKF (N, M, H, IH, RI, C, RR, V, IV, B, IB, U, W, IFAIL) | On-line doc: CALL GAMSDOC F02BKF (or @PRT NAG*DOC.F02BKF) | Access: LIB NBS*NAG | See also: F02AKF
- F02BLE** Compute selected eigenvectors of a complex upper Hessenberg matrix by inverse iteration, given estimates of their associated eigenvalues. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02BLF. | Class(es): D4c3 | Usage: CALL F02BLE (N, M, HR, IHR, HI, IHI, RI, C, RR, VR, IVR, VI, IVI, BR, IBR, BI, IBI, U, W, IFAIL) | On-line doc: CALL GAMSDOC F02BLE (or @PRT NAG*DOC.F02BLE) | Access: LIB NBS*NAG | See also: F02AKE
- F02BLF** Compute selected eigenvectors of a complex upper Hessenberg matrix by inverse iteration, given estimates of their associated

eigenvalues. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02BLE. | Class(es): D4c3 | Usage: CALL F02BLF (N, M, HR, IHR, HI, IHI, RI, C, RR, VR, IVR, VI, IVI, BR, IBR, BI, IBI, U, W, IFAIL) | On-line doc: CALL GAMSDOC F02BLF (or @PRT NAG*DOC.F02BLF) | Access: LIB NBS*NAG | See also: F02AKF

F02GJE Generalised eigenproblem $Ax=kBx$, QZ algorithm (black box), complex matrices, alleigenvalues and (optionally) eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02GJF. | Class(es): D4b4 | Usage: CALL F02GJE (N, AR, IAR, AI, IAI, BR, IBR, BI, IBI, EPS1, ALFR, ALFI, BETA, MATV VR, IVR, VI, IVI, ITER, IFAIL) | On-line doc: CALL GAMSDOC F02GJE (or @PRT NAG*DOC.F02GJE) | Access: LIB NBS*NAG

F02GJF Generalised eigenproblem $Ax=kBx$, QZ algorithm (black box), complex matrices, alleigenvalues and (optionally) eigenvectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02GJE. | Class(es): D4b4 | Usage: CALL F02GJF (N, AR, IAR, AI, IAI, BR, IBR, BI, IBI, EPS1, ALFR, ALFI, BETA, MATV VR, IVR, VI, IVI, ITER, IFAIL) | On-line doc: CALL GAMSDOC F02GJF (or @PRT NAG*DOC.F02GJF) | Access: LIB NBS*NAG

F02SDE Generalised real eigenproblem $Ax=kBx$, where A and B are band matrices, eigenvector by inverse iteration. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02SDF. | Class(es): D4b5 | Usage: CALL F02SDE (N, MA1, MB1, A, IA, B, IB, SYM, RELEP, RMU, VEC, D, INT, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02SDE (or @PRT NAG*DOC.F02SDE) | Access: LIB NBS*NAG

F02SDF Generalised real eigenproblem $Ax=kBx$, where A and B are band matrices, eigenvector by inverse iteration. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02SDE. | Class(es): D4b5 | Usage: CALL F02SDF (N, MA1, MB1, A, IA, B, IB, SYM, RELEP, RMU, VEC, D, INT, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02SDF (or @PRT NAG*DOC.F02SDF) | Access: LIB NBS*NAG

F02SZE Singular value decomposition of a real bidiagonal matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02SZF. | Class(es): D6 | Usage: CALL F02SZE (N, D, E, SV, WANTB, B, WANTY, Y, NRY, LY, WANTZ, Z, NRZ, NCZ, WORK1 WORK2, WORK3, IFAIL) | On-line doc: CALL GAMSDOC F02SZE (or @PRT NAG*DOC.F02SZE) | Access: LIB NBS*NAG

F02SZF Singular value decomposition of a real bidiagonal matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02SZE. | Class(es): D6 | Usage: CALL F02SZF (N, D, E, SV, WANTB, B, WANTY, Y, NRY, LY, WANTZ, Z, NRZ, NCZ, WORK1 WORK2, WORK3, IFAIL) | On-line doc: CALL GAMSDOC F02SZF (or @PRT NAG*DOC.F02SZF) | Access: LIB NBS*NAG

F02WAE Singular value decomposition of a real $m \times n$ matrix, singular values and right singular vectors, ($m \geq n$). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02WAF. | Class(es): D6 | Usage: CALL F02WAE (M, N, A, NRA, WANTB, B, SV, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02WAE (or @PRT NAG*DOC.F02WAE) | Access: LIB NBS*NAG

F02WAF Singular value decomposition of a real $m \times n$ matrix, singular values and right singular vectors, ($m \geq n$). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02WAE. | Class(es): D6 | Usage: CALL F02WAF (M, N, A, NRA, WANTB, B, SV, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02WAF (or @PRT NAG*DOC.F02WAF) | Access: LIB NBS*NAG

F02WBE Singular value decomposition of a real $m \times n$ matrix, singular values and right singular vectors, ($m \leq n$). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02WBF. | Class(es): D6 | Usage: CALL F02WBE (M, N, A, NRA, WANTB, B, SV, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02WBE (or @PRT NAG*DOC.F02WBE) | Access: LIB NBS*NAG

F02WBF Singular value decomposition of a real $m \times n$ matrix, singular values and right singular vectors, ($m \leq n$). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02WBE. | Class(es): D6 | Usage: CALL F02WBF (M, N, A, NRA, WANTB, B, SV, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02WBF (or @PRT NAG*DOC.F02WBF) | Access: LIB NBS*NAG

F02WCE Singular value decomposition of a real $m \times n$ matrix, singular values and left and right singular vectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02WCF. | Class(es): D6 | Usage: CALL F02WCE (M, N, MINMN, A, NRA, Q, NRQ, SV, PT, NRPT, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02WCE (or @PRT NAG*DOC.F02WCE) | Access: LIB NBS*NAG

F02WCF Singular value decomposition of a real $m \times n$ matrix, singular values and left and right singular vectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02WCE. | Class(es): D6 | Usage: CALL F02WCF (M, N, MINMN, A, NRA, Q, NRQ, SV, PT, NRPT, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02WCF (or @PRT NAG*DOC.F02WCF) | Access: LIB NBS*NAG

F02WDE S.V.D. of a real $m \times n$ matrix, singular values and (optionally) right singular vectors, optionally or conditionally following QR-factorisation ($m \geq n$). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02WDF. | Class(es): D5 D6 | Usage: CALL F02WDE (M, N, A, NRA, WANTB, B, TOL, SVD, IRANK, Z, SV, WANTR, R, NRR, WANTPT, PT, NRPT, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02WDE (or @PRT NAG*DOC.F02WDE) | Access: LIB NBS*NAG

F02WDF S.V.D. of a real $m \times n$ matrix, singular values and (optionally) right singular vectors, optionally or conditionally following QR-factorisation ($m \geq n$). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02WDE. | Class(es): D5 D6 | Usage: CALL F02WDF (M, N, A, NRA, WANTB, B, TOL, SVD, IRANK, Z, SV, WANTR, R, NRR, WANTPT, PT, NRPT, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02WDF (or @PRT NAG*DOC.F02WDF) | Access: LIB NBS*NAG

F03AAE Determinant (black box), real matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F03AAF. | Class(es): D3a1 | Usage: CALL F03AAE (A, IA, N, DET, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F03AAE (or

@PRT NAG*DOC.F03AAE) | Access: LIB NBS*NAG

F03AAF Determinant (black box), real matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F03AAE. | Class(es): D3a1 | Usage: CALL F03AAF (A, IA, N, DET, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F03AAF (or @PRT NAG*DOC.F03AAF) | Access: LIB NBS*NAG

F03ABE Determinant (black box), real symmetric positive-definite matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F03ABF. | Class(es): D3b1b | Usage: CALL F03ABE (A, IA, N, DET, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F03ABE (or @PRT NAG*DOC.F03ABE) | Access: LIB NBS*NAG

F03ABF Determinant (black box), real symmetric positive-definite matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F03ABE. | Class(es): D3b1b | Usage: CALL F03ABF (A, IA, N, DET, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F03ABF (or @PRT NAG*DOC.F03ABF) | Access: LIB NBS*NAG

F03ACE Determinant (black box), real symmetric positive-definite band matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F03ACF. | Class(es): D3b2 | Usage: CALL F03ACE (A, IA, N, M, DET, RL, IL, M1, IFAIL) | On-line doc: CALL GAMSDOC F03ACE (or @PRT NAG*DOC.F03ACE) | Access: LIB NBS*NAG

F03ACF Determinant (black box), real symmetric positive-definite band matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F03ACE. | Class(es): D3b2 | Usage: CALL F03ACF (A, IA, N, M, DET, RL, IL, M1, IFAIL) | On-line doc: CALL GAMSDOC F03ACF (or @PRT NAG*DOC.F03ACF) | Access: LIB NBS*NAG

F03ADE Determinant (black box), complex matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F03ADF. | Class(es): D3c1 | Usage: CALL F03ADE (A, IA, N, DETR, DETI, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F03ADE (or @PRT NAG*DOC.F03ADE) | Access: LIB NBS*NAG

F03ADF Determinant (black box), complex matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F03ADE. | Class(es): D3c1 | Usage: CALL F03ADF (A, IA, N, DETR, DETI, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F03ADF (or @PRT NAG*DOC.F03ADF) | Access: LIB NBS*NAG

F03AEE LLT-factorisation and determinant, real symmetric positive-definite matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F03AEF. | Class(es): D2b1b D3b1b | Usage: CALL F03AEE (N, A, IA, P, D1, ID, IFAIL) | On-line doc: CALL GAMSDOC F03AEE (or @PRT NAG*DOC.F03AEE) | Access: LIB NBS*NAG

F03AEF LLT-factorisation and determinant, real symmetric positive-definite matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F03AEE. | Class(es): D2b1b D3b1b | Usage: CALL F03AEF (N, A, IA, P, D1, ID, IFAIL) | On-line doc: CALL GAMSDOC F03AEF (or @PRT NAG*DOC.F03AEF) | Access: LIB NBS*NAG

F03AFE LU-factorisation and determinant, real matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F03AFF. | Class(es): D2a1 D3a1 | Usage: CALL F03AFE (N, EPS, A, IA, D1, ID, P, IFAIL) | On-line doc: CALL GAMSDOC F03AFE (or @PRT NAG*DOC.F03AFE) | Access: LIB NBS*NAG

F03AFF LU-factorisation and determinant, real matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F03AFE. | Class(es): D2a1 D3a1 | Usage: CALL F03AFF (N, EPS, A, IA, D1, ID, P, IFAIL) | On-line doc: CALL GAMSDOC F03AFF (or @PRT NAG*DOC.F03AFF) | Access: LIB NBS*NAG

F03AGE LLT-factorisation and determinant, real symmetric positive-definite band matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F03AGF. | Class(es): D2b2 D3b2 | Usage: CALL F03AGE (N, M, A, IA, RL, IL, M1, D1, ID, IFAIL) | On-line doc: CALL GAMSDOC F03AGE (or @PRT NAG*DOC.F03AGE) | Access: LIB NBS*NAG

F03AGF LLT-factorisation and determinant, real symmetric positive-definite band matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F03AGE. | Class(es): D2b2 D3b2 | Usage: CALL F03AGF (N, M, A, IA, RL, IL, M1, D1, ID, IFAIL) | On-line doc: CALL GAMSDOC F03AGF (or @PRT NAG*DOC.F03AGF) | Access: LIB NBS*NAG

F03AHE LU-factorisation and determinant, complex matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F03AHF. | Class(es): D2c1 D3c1 | Usage: CALL F03AHE (N, A, IA, DETR, DETI, ID, RINT, IFAIL) | On-line doc: CALL GAMSDOC F03AHE (or @PRT NAG*DOC.F03AHE) | Access: LIB NBS*NAG

F03AHF LU-factorisation and determinant, complex matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F03AHE. | Class(es): D2c1 D3c1 | Usage: CALL F03AHF (N, A, IA, DETR, DETI, ID, RINT, IFAIL) | On-line doc: CALL GAMSDOC F03AHF (or @PRT NAG*DOC.F03AHF) | Access: LIB NBS*NAG

F03AME Determinant of a complex Hermitian positive-definite matrix, after factorisation by F01BNE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F03AMF. | Class(es): D3d1b | Usage: CALL F03AME (N, TEN, P, D1, D2) | On-line doc: CALL GAMSDOC F03AME (or @PRT NAG*DOC.F03AME) | Access: LIB NBS*NAG | See also: F01BNE

F03AMF Determinant of a complex Hermitian positive-definite matrix, after factorisation by F01BNF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F03AME. | Class(es): D3d1b | Usage: CALL F03AMF (N, TEN, P, D1, D2) | On-line doc: CALL GAMSDOC F03AMF (or @PRT NAG*DOC.F03AMF) | Access: LIB NBS*NAG | See also: F01BNF

F04AAE Simultaneous linear equations (black box), real matrix, approximate solution, multiple right hand sides. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AAF. | Class(es): D2a1 | Usage: CALL F04AAE (A, IA, B, IB, N, M, C, IC, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F04AAE (or @PRT NAG*DOC.F04AAE) | Access: LIB NBS*NAG

- F04AAF** Simultaneous linear equations (black box), real matrix, approximate solution, multiple right hand sides. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AAE. | Class(es): D2a1 | Usage: CALL F04AAF (A, IA, B, IB, N, M, C, IC, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F04AAF (or @PRT NAG*DOC.F04AAF) | Access: LIB NBS*NAG
- F04ABE** Simultaneous linear equations (black box), real symmetric positive-definite matrix, accurate solution, multiple right hand sides. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04ABF. | Class(es): D2b1b | Usage: CALL F04ABE (A, IA, B, IB, N, M, C, IC, WKSPCE, BB, IBB, IFAIL) | On-line doc: CALL GAMSDOC F04ABE (or @PRT NAG*DOC.F04ABE) | Access: LIB NBS*NAG
- F04ABF** Simultaneous linear equations (black box), real symmetric positive-definite matrix, accurate solution, multiple right hand sides. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04ABE. | Class(es): D2b1b | Usage: CALL F04ABF (A, IA, B, IB, N, M, C, IC, WKSPCE, BB, IBB, IFAIL) | On-line doc: CALL GAMSDOC F04ABF (or @PRT NAG*DOC.F04ABF) | Access: LIB NBS*NAG
- F04ACE** Simultaneous linear equations (black box), real symmetric positive-definite bandmatrix, approximate solution, multiple right hand sides. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04ACF. | Class(es): D2b2 | Usage: CALL F04ACE (A, IA, B, IB, N, M, IR, C, IC, RL, IRL, M1, IFAIL) | On-line doc: CALL GAMSDOC F04ACE (or @PRT NAG*DOC.F04ACE) | Access: LIB NBS*NAG
- F04ACF** Simultaneous linear equations (black box), real symmetric positive-definite bandmatrix, approximate solution, multiple right hand sides. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04ACE. | Class(es): D2b2 | Usage: CALL F04ACF (A, IA, B, IB, N, M, IR, C, IC, RL, IRL, M1, IFAIL) | On-line doc: CALL GAMSDOC F04ACF (or @PRT NAG*DOC.F04ACF) | Access: LIB NBS*NAG
- F04ADE** Simultaneous linear equations (black box), complex matrix, approximate solution, multiple right hand sides. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04ADF. | Class(es): D2c1 | Usage: CALL F04ADE (A, IA, B, IB, N, M, C, IC, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F04ADE (or @PRT NAG*DOC.F04ADE) | Access: LIB NBS*NAG
- F04ADF** Simultaneous linear equations (black box), complex matrix, approximate solution, multiple right hand sides. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04ADE. | Class(es): D2c1 | Usage: CALL F04ADF (A, IA, B, IB, N, M, C, IC, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F04ADF (or @PRT NAG*DOC.F04ADF) | Access: LIB NBS*NAG
- F04AEE** Simultaneous linear equations (black box), real matrix, accurate solution, multiple right hand sides. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AEF. | Class(es): D2a1 | Usage: CALL F04AEE (A, IA, B, IB, N, M, C, IC, WKSPCE, AA, IAA, BB, IBB, IFAIL) | On-line doc: CALL GAMSDOC F04AEE (or @PRT NAG*DOC.F04AEE) | Access: LIB NBS*NAG
- F04AEF** Simultaneous linear equations (black box), real matrix, accurate solution, multiple right hand sides. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AEE. | Class(es): D2a1 | Usage: CALL F04AEF (A, IA, B, IB, N, M, C, IC, WKSPCE, AA, IAA, BB, IBB, IFAIL) | On-line doc: CALL GAMSDOC F04AEF (or @PRT NAG*DOC.F04AEF) | Access: LIB NBS*NAG
- F04AFE** Calculates the accurate solution of a set of real symmetric positive definite linear equations with multiple right hand sides, $AX = B$, where A has been decomposed into triangular matrices using F03AEE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AFF. | Class(es): D2b1b | Usage: CALL F04AFE (N, IR, A, IA, P, B, IB, EPS, X, IX, BB, IBB, K, IFAIL) | On-line doc: CALL GAMSDOC F04AFE (or @PRT NAG*DOC.F04AFE) | Access: LIB NBS*NAG | See also: F03AEE
- F04AFF** Calculates the accurate solution of a set of real symmetric positive definite linear equations with multiple right hand sides, $AX = B$, where A has been decomposed into triangular matrices using F03AEF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AFE. | Class(es): D2b1b | Usage: CALL F04AFF (N, IR, A, IA, P, B, IB, EPS, X, IX, BB, IBB, K, IFAIL) | On-line doc: CALL GAMSDOC F04AFF (or @PRT NAG*DOC.F04AFF) | Access: LIB NBS*NAG | See also: F03AEF
- F04AGE** Calculates the approximate solution of a set of real symmetric positive definite linear equations with multiple right hand sides, $AX = B$, where A has been decomposed into triangular matrices using F03AEE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AGF. | Class(es): D2b1b | Usage: CALL F04AGE (N, IR, A, IA, P, B, IB, X, IX) | On-line doc: CALL GAMSDOC F04AGE (or @PRT NAG*DOC.F04AGE) | Access: LIB NBS*NAG | See also: F03AEE
- F04AGF** Calculates the approximate solution of a set of real symmetric positive definite linear equations with multiple right hand sides, $AX = B$, where A has been decomposed into triangular matrices using F03AEF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AGE. | Class(es): D2b1b | Usage: CALL F04AGF (N, IR, A, IA, P, B, IB, X, IX) | On-line doc: CALL GAMSDOC F04AGF (or @PRT NAG*DOC.F04AGF) | Access: LIB NBS*NAG | See also: F03AEF
- F04AHE** Calculates the accurate solution of set of real linear equations with multiple right hand sides, $AX=B$, where A has been decomposed into triangular matrices using F03AFE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AHF. | Class(es): D2a1 | Usage: CALL F04AHE (N, IR, A, IA, AA, IAA, P, B, IB, EPS, X, IX, BB, IBB, K, IFAIL) | On-line doc: CALL GAMSDOC F04AHE (or @PRT NAG*DOC.F04AHE) | Access: LIB NBS*NAG | See also: F03AFE
- F04AHF** Calculates the accurate solution of set of real linear equations with multiple right hand sides, $AX=B$, where A has been decomposed into triangular matrices using F03AFF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AHE. | Class(es): D2a1 | Usage: CALL F04AHF (N, IR, A, IA, AA, IAA, P, B, IB, EPS, X, IX, BB, IBB, K, IFAIL) | On-line doc: CALL GAMSDOC F04AHF (or @PRT NAG*DOC.F04AHF) | Access: LIB NBS*NAG | See also: F03AFF
- F04AJE** Calculates the approximate solution of set of real linear equations with multiple right hand sides, $AX=B$, where A has been decomposed into triangular matrices using F03AFE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AJF. | Class(es): D2a1 | Usage: CALL F04AJE (N, IR, A, IA, P, B, IB) | On-line doc: CALL GAMSDOC F04AJE (or @PRT NAG*DOC.F04AJE) | Access: LIB NBS*NAG | See also: F03AFE

- F04AJF** Calculates the approximate solution of set of real linear equations with multiple right hand sides, $AX=B$, where A has been decomposed into triangular matrices using F03AFF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AJE. | Class(es): D2a1 | Usage: CALL F04AJF (N, IR, A, IA, P, B, IB) | On-line doc: CALL GAMSDOC F04AJF (or @PRT NAG*DOC.F04AJF) | Access: LIB NBS*NAG | See also: F03AFF
- F04AKE** Calculates the approximate solution of a set of complex linear equations with multiple right hand sides, $AX = B$, where A has been decomposed into triangular matrices using F03AHE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AKF. | Class(es): D2c1 | Usage: CALL F04AKE (N, IR, A, IA, P, B, IB) | On-line doc: CALL GAMSDOC F04AKE (or @PRT NAG*DOC.F04AKE) | Access: LIB NBS*NAG | See also: F03AHE
- F04AKF** Calculates the approximate solution of a set of complex linear equations with multiple right hand sides, $AX = B$, where A has been decomposed into triangular matrices using F03AHF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AKE. | Class(es): D2c1 | Usage: CALL F04AKF (N, IR, A, IA, P, B, IB) | On-line doc: CALL GAMSDOC F04AKF (or @PRT NAG*DOC.F04AKF) | Access: LIB NBS*NAG | See also: F03AHF
- F04ALE** Calculates the approximate solution of a set of real symmetric positive definite band linear equations with multiple right hand sides, $AX = B$, where A has been decomposed into triangular matrices using F03AGE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04ALF. | Class(es): D2b2 | Usage: CALL F04ALE (N, M, IR, RL, IRL, M1, B, IB, X, IX) | On-line doc: CALL GAMSDOC F04ALE (or @PRT NAG*DOC.F04ALE) | Access: LIB NBS*NAG | See also: F03AGE
- F04ALF** Calculates the approximate solution of a set of real symmetric positive definite band linear equations with multiple right hand sides, $AX = B$, where A has been decomposed into triangular matrices using F03AGF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04ALE. | Class(es): D2b2 | Usage: CALL F04ALF (N, M, IR, RL, IRL, M1, B, IB, X, IX) | On-line doc: CALL GAMSDOC F04ALF (or @PRT NAG*DOC.F04ALF) | Access: LIB NBS*NAG | See also: F03AGF
- F04AME** Least-squares, m real equations in n unknowns, rank =n, $m \geq n$, accurate solution (black box). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AMF. | Class(es): D9 | Usage: CALL F04AME (A, IA, X, IX, B, IB, M, N, IR, EPS, QR, IQR, ALPHA, E, Y, Z, R, IPIV, IFAIL) | On-line doc: CALL GAMSDOC F04AME (or @PRT NAG*DOC.F04AME) | Access: LIB NBS*NAG
- F04AMF** Least-squares, m real equations in n unknowns, rank =n, $m \geq n$, accurate solution (black box). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AME. | Class(es): D9 | Usage: CALL F04AMF (A, IA, X, IX, B, IB, M, N, IR, EPS, QR, IQR, ALPHA, E, Y, Z, R, IPIV, IFAIL) | On-line doc: CALL GAMSDOC F04AMF (or @PRT NAG*DOC.F04AMF) | Access: LIB NBS*NAG
- F04ANE** Least-squares, m real equations in n unknowns, rank =n, $m \geq n$, approximate solution (after factorisation by F01AXE). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04ANF. | Class(es): D9 | Usage: CALL F04ANE (M, N, QR, IQR, ALPHA, IPIV, B, X, Z) | On-line doc: CALL GAMSDOC F04ANE (or @PRT NAG*DOC.F04ANE) | Access: LIB NBS*NAG | See also: F01AXE
- F04ANF** Least-squares, m real equations in n unknowns, rank =n, $m \geq n$, approximate solution (after factorisation by F01AXF). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04ANE. | Class(es): D9 | Usage: CALL F04ANF (M, N, QR, IQR, ALPHA, IPIV, B, X, Z) | On-line doc: CALL GAMSDOC F04ANF (or @PRT NAG*DOC.F04ANF) | Access: LIB NBS*NAG | See also: F01AXF
- F04AQE** Calculates the approximate solution of a set of real symmetric positive definite linear equations with a single right hand side, $Ax = b$, where A has been decomposed into LDLt using F01BQE. (Economical storage.). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AQF. | Class(es): D2b1b | Usage: CALL F04AQE (N, M, RL, D, B, X) | On-line doc: CALL GAMSDOC F04AQE (or @PRT NAG*DOC.F04AQE) | Access: LIB NBS*NAG | See also: F01BQF
- F04AQF** Calculates the approximate solution of a set of real symmetric positive definite linear equations with a single right hand side, $Ax = b$, where A has been decomposed into LDLt using F01BQF. (Economical storage.). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AQE. | Class(es): D2b1b | Usage: CALL F04AQF (N, M, RL, D, B, X) | On-line doc: CALL GAMSDOC F04AQF (or @PRT NAG*DOC.F04AQF) | Access: LIB NBS*NAG | See also: F01BQF
- F04ARE** Simultaneous linear equations (black box), real matrix, approximate solution, one right hand side. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04ARF. | Class(es): D2a1 | Usage: CALL F04ARE (A, IA, B, N, C, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F04ARE (or @PRT NAG*DOC.F04ARE) | Access: LIB NBS*NAG
- F04ARF** Simultaneous linear equations (black box), real matrix, approximate solution, one right hand side. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04ARE. | Class(es): D2a1 | Usage: CALL F04ARF (A, IA, B, N, C, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F04ARF (or @PRT NAG*DOC.F04ARF) | Access: LIB NBS*NAG
- F04ASE** Simultaneous linear equations (black box), real symmetric positive-definite matrix, accurate solution, one right hand side. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04ASF. | Class(es): D2b1b | Usage: CALL F04ASE (A, IA, B, N, C, WK1, WK2, IFAIL) | On-line doc: CALL GAMSDOC F04ASE (or @PRT NAG*DOC.F04ASE) | Access: LIB NBS*NAG
- F04ASF** Simultaneous linear equations (black box), real symmetric positive-definite matrix, accurate solution, one right hand side. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04ASE. | Class(es): D2b1b | Usage: CALL F04ASF (A, IA, B, N, C, WK1, WK2, IFAIL) | On-line doc: CALL GAMSDOC F04ASF (or @PRT NAG*DOC.F04ASF) | Access: LIB NBS*NAG
- F04ATE** Simultaneous linear equations (black box), real matrix, accurate solution, one right hand side. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04ATF. | Class(es): D2a1 | Usage: CALL F04ATE (A, IA, B, N, C, AA, IAA,

WKS1, WKS2, IFAIL) | On-line doc: CALL GAMSDOC F04ATE (or @PRT NAG*DOC.F04ATE) | Access: LIB NBS*NAG

F04ATF Simultaneous linear equations (black box), real matrix, accurate solution, one right hand side. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04ATE. | Class(es): D2a1 | Usage: CALL F04ATF (A, IA, B, N, C, AA, IAA, WKS1, WKS2, IFAIL) | On-line doc: CALL GAMSDOC F04ATF (or @PRT NAG*DOC.F04ATF) | Access: LIB NBS*NAG

F04AWE Calculates the approximate solutions of a set of complex linear equations with multiple right hand sides, $AX = B$, where A is positive definite Hermitian, following the Cholesky decomposition of A by F01BNE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AWF. | Class(es): D2d1b | Usage: CALL F04AWE (N, IR, A, IA, P, B, IB, X, IX) | On-line doc: CALL GAMSDOC F04AWE (or @PRT NAG*DOC.F04AWE) | Access: LIB NBS*NAG | See also: F01BNE

F04AWF Calculates the approximate solutions of a set of complex linear equations with multiple right hand sides, $AX = B$, where A is positive definite Hermitian, following the Cholesky decomposition of A by F01BNF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AWE. | Class(es): D2d1b | Usage: CALL F04AWF (N, IR, A, IA, P, B, IB, X, IX) | On-line doc: CALL GAMSDOC F04AWF (or @PRT NAG*DOC.F04AWF) | Access: LIB NBS*NAG | See also: F01BNF

F04AXE Calculates the approximate solution of a set of real sparse linear equations with a single right hand side, $Ax = b$ or $Atx = b$, where A has been decomposed by F01BRE or F01BSE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AXF. | Class(es): D2a4 | Usage: CALL F04AXE (N, A, LICN, ICN, IKEEP, RHS, W, MTYPE, IDISP, RESID) | On-line doc: CALL GAMSDOC F04AXE (or @PRT NAG*DOC.F04AXE) | Access: LIB NBS*NAG | See also: F01BRE F01BSE

F04AXF Calculates the approximate solution of a set of real sparse linear equations with a single right hand side, $Ax = b$ or $Atx = b$, where A has been decomposed by F01BRF or F01BSF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AXE. | Class(es): D2a4 | Usage: CALL F04AXF (N, A, LICN, ICN, IKEEP, RHS, W, MTYPE, IDISP, RESID) | On-line doc: CALL GAMSDOC F04AXF (or @PRT NAG*DOC.F04AXF) | Access: LIB NBS*NAG | See also: F01BRF F01BSF

F04AYE Calculates the approximate solution of set of real linear equations with multiple right hand sides, $AX=B$, where A has been decomposed into triangular matrices using F01BTE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AYF. | Class(es): D2a1 | Usage: CALL F04AYE (N, IR, A, IA, P, B, IB, IFAIL) | On-line doc: CALL GAMSDOC F04AYE (or @PRT NAG*DOC.F04AYE) | Access: LIB NBS*NAG | See also: F01BTE

F04AYF Calculates the approximate solution of set of real linear equations with multiple right hand sides, $AX=B$, where A has been decomposed into triangular matrices using F01BTF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AYE. | Class(es): D2a1 | Usage: CALL F04AYF (N, IR, A, IA, P, B, IB, IFAIL) | On-line doc: CALL GAMSDOC F04AYF (or @PRT NAG*DOC.F04AYF) | Access: LIB NBS*NAG | See also: F01BTF

F04AZE Calculates the approximate solution of a set of real symmetric positive definite linear equations with a single right hand side, $Ax = b$, where A has been decomposed into triangular matrices using F01BXE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AZF. | Class(es): D2b1b | Usage: CALL F04AZE (N, IR, A, IA, P, B, IB, IFAIL) | On-line doc: CALL GAMSDOC F04AZE (or @PRT NAG*DOC.F04AZE) | Access: LIB NBS*NAG | See also: F01BXE

F04AZF Calculates the approximate solution of a set of real symmetric positive definite linear equations with a single right hand side, $Ax = b$, where A has been decomposed into triangular matrices using F01BXF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AZE. | Class(es): D2b1b | Usage: CALL F04AZF (N, IR, A, IA, P, B, IB, IFAIL) | On-line doc: CALL GAMSDOC F04AZF (or @PRT NAG*DOC.F04AZF) | Access: LIB NBS*NAG | See also: F01BXF

F04JAE Least-squares, m real equations in n unknowns, $\text{rank} \leq n$, $m \geq n$, minimal least-squares solution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04JAF. | Class(es): D9 | Usage: CALL F04JAE (M, N, A, NRA, B, TOL, SIGMA, IRANK, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F04JAE (or @PRT NAG*DOC.F04JAE) | Access: LIB NBS*NAG

F04JAF Least-squares, m real equations in n unknowns, $\text{rank} \leq n$, $m \geq n$, minimal least-squares solution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04JAE. | Class(es): D9 | Usage: CALL F04JAF (M, N, A, NRA, B, TOL, SIGMA, IRANK, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F04JAF (or @PRT NAG*DOC.F04JAF) | Access: LIB NBS*NAG

F04JDE Least-squares, m real equations in n unknowns, $\text{rank} \leq n$, $m \leq n$, minimal least-squares solution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04JDF. | Class(es): D9 | Usage: CALL F04JDE (M, N, A, NRA, B, TOL, SIGMA, IRANK, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F04JDE (or @PRT NAG*DOC.F04JDE) | Access: LIB NBS*NAG

F04JDF Least-squares, m real equations in n unknowns, $\text{rank} \leq n$, $m \leq n$, minimal least-squares solution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04JDE. | Class(es): D9 | Usage: CALL F04JDF (M, N, A, NRA, B, TOL, SIGMA, IRANK, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F04JDF (or @PRT NAG*DOC.F04JDF) | Access: LIB NBS*NAG

F04JGE Least-squares, m real equations in n unknowns, $\text{rank} \leq n$, $m \geq n$, least-squares solution if $\text{rank} = n$, otherwise minimal least-squares solution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04JGF. | Class(es): D9 | Usage: CALL F04JGE (M, N, A, NRA, B, TOL, SVD, SIGMA, IRANK, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F04JGE (or @PRT NAG*DOC.F04JGE) | Access: LIB NBS*NAG

F04JGF Least-squares, m real equations in n unknowns, $\text{rank} \leq n$, $m \geq n$, least-squares solution if $\text{rank} = n$, otherwise minimal least-squares solution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04JGE. | Class(es): D9 | Usage: CALL F04JGF (M, N, A, NRA, B, TOL, SVD, SIGMA, IRANK, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F04JGF (or @PRT NAG*DOC.F04JGF) | Access: LIB NBS*NAG

F04LDE Simultaneous linear equations (factorising the matrix of coefficients), real band matrix, approximate solution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04LDF. | Class(es): D2a2 | Usage: CALL F04LDE (N, M1,

M2, IR, A, IA, AL, IL, IN, B, IB, IFAIL) | On-line doc: CALL GAMSDOC F04LDE (or @PRT NAG*DOC.F04LDE) | Access: LIB NBS*NAG | See also: F01LBE

F04LDF Simultaneous linear equations (factorising the matrix of coefficients), real band matrix, approximate solution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04LDE. | Class(es): D2a2 | Usage: CALL F04LDF (N, M1, M2, IR, A, IA, AL, IL, IN, B, IB, IFAIL) | On-line doc: CALL GAMSDOC F04LDF (or @PRT NAG*DOC.F04LDF) | Access: LIB NBS*NAG | See also: F01LBF

F04MCE Calculates the approximate solution of a system of real linear equations with multiple right hand sides, $Ax = B$, where A is a symmetric positive definite variable-bandwidth matrix, which has previously been factorised by F01MCE. Related systems may also be solved. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04MCF. | Class(es): D2b2 | Usage: CALL F04MCE (N, AL, LAL, D, NROW, IR, B, NRB, ISELCT, X, NRX, IFAIL) | On-line doc: CALL GAMSDOC F04MCE (or @PRT NAG*DOC.F04MCE) | Access: LIB NBS*NAG | See also: F01MCE

F04MCF Calculates the approximate solution of a system of real linear equations with multiple right hand sides, $Ax = B$, where A is a symmetric positive definite variable-bandwidth matrix, which has previously been factorised by F01MCF. Related systems may also be solved. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04MCE. | Class(es): D2b2 | Usage: CALL F04MCF (N, AL, LAL, D, NROW, IR, B, NRB, ISELCT, X, NRX, IFAIL) | On-line doc: CALL GAMSDOC F04MCF (or @PRT NAG*DOC.F04MCF) | Access: LIB NBS*NAG | See also: F01MCF

F05AAE Schmidt orthogonalisation of n vectors of order m. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F05AAF. | Class(es): D5 | Usage: CALL F05AAE (A, IA, M, N1, N2, S, CC, ICOL, IFAIL) | On-line doc: CALL GAMSDOC F05AAE (or @PRT NAG*DOC.F05AAE) | Access: LIB NBS*NAG

F05AAF Schmidt orthogonalisation of n vectors of order m. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F05AAE. | Class(es): D5 | Usage: CALL F05AAF (A, IA, M, N1, N2, S, CC, ICOL, IFAIL) | On-line doc: CALL GAMSDOC F05AAF (or @PRT NAG*DOC.F05AAF) | Access: LIB NBS*NAG

F05ABE Approximate 2-norm of a vector. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F05ABF. | Class(es): D1a3b | Usage: D = F05ABE (X, N) | On-line doc: CALL GAMSDOC F05ABE (or @PRT NAG*DOC.F05ABE) | Access: LIB NBS*NAG

F05ABF Approximate 2-norm of a vector. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F05ABE. | Class(es): D1a3b | Usage: D = F05ABF (X, N) | On-line doc: CALL GAMSDOC F05ABF (or @PRT NAG*DOC.F05ABF) | Access: LIB NBS*NAG

FAC Factorial, $= n!$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DFAC. | Class(es): C1 | Usage: I = FAC (N) | On-line doc: CALL GAMSDOC FAC (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

FC Fits piecewise polynomial to discrete data with equality and inequality constraints. | Portable single precision Fortran subprogram in FC sublibrary of CMLIB library. | Class(es): K1a1a1 K1a2a L8a3 | Usage: CALL FC(NDATA,XDATA,YDATA,SDATA,NORD,NBKPT,BKPT,NCONST,XCONST,YCONST,NDERIV,MODE,COEFF,W,IW) | On-line doc: CALL GAMSDOC FC (or @PRT CMLIB*DOC.FC/FC) | Tests: CMLIB*TEST-SOURCE.\$F/FC | Access: LIB NBS*CMLIB | See also: CV

FCDF Computes the cumulative distribution function value for the F-distribution with degrees of freedom parameters NU1 and NU2. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1f | Usage: CALL FCDF(X,NU1,NU2,CDF) | On-line doc: CALL GAMSDOC FCDF (or @PRT DATAPAC*DOC.FCDF) | Access: LIB NBS*DATAPAC

FFT Compute FFT of complex data sequence (forward or inverse) any number of points. Useful for multivariate transforms. Uses only real arithmetic. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DFFT. | Class(es): J1a2 J1b | Usage: CALL FFT (A,B,NTOT,N,NSPAN,ISN) | On-line doc: CALL GAMSDOC FFT (or @PRT PORT*DOC.FFT) | Access: LIB NBS*PORT

FFT2C Computes the fast Fourier transform of a complex valued sequence of length equal to a power two. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): J1a2 | Usage: CALL FFT2C (A,M,IWK) | On-line doc: CALL GAMSDOC FFT2C (or @PRT IMSL*DOC.FFT2C) | Access: LIB NBS*IMSL

FFT3D Compute the fast Fourier transform of a complex valued 1,2 or 3 dimensional array. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): J1b | Usage: CALL FFT3D (A,IA1,IA2,N1,N2,N3,IJOB,IWK,RWK,CWK) | On-line doc: CALL GAMSDOC FFT3D (or @PRT IMSL*DOC.FFT3D) | Access: LIB NBS*IMSL

FFTC Mixed radix fast Fourier transform of complex data. Two arrays used for complex data. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DFFTC. | Class(es): J1a2 | Usage: CALL FFTC (N,AR,AI) | On-line doc: CALL GAMSDOC FFTC (or @PRT PORT*DOC.FFTC) | Access: LIB NBS*PORT | See also: FFTCI

FFTCC Compute the fast Fourier transform of a complex valued sequence. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): J1a2 | Usage: CALL FFTCC (A,N,IWK,WK) | On-line doc: CALL GAMSDOC FFTCC (or @PRT IMSL*DOC.FFTCC) | Access: LIB NBS*IMSL

FFTCI Finds the inverse fast Fourier transform, given the Fourier coefficients in the frequency domain. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DFFTCI. | Class(es): J1a2 | Usage: CALL FFTCI (N,FR,FI) | On-line doc: CALL GAMSDOC FFTCI (or @PRT PORT*DOC.FFTCI) | Access: LIB NBS*PORT | See also: FFTC

FFTR Mixed radix fast Fourier transform to find the transform of 2N real data points. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DFFTR. | Class(es): J1a1 | Usage: CALL FFTR (NNP2,A,B) | On-line doc: CALL GAMSDOC FFTR (or @PRT PORT*DOC.FFTR) | Access: LIB NBS*PORT | See also: FFTRI

- FFTRC** Compute the fast Fourier transform of a real valued sequence. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): J1a1 | Usage: CALL FFTRC (A,N,X,IWK,WK) | On-line doc: CALL GAMSDOC FFTRC (or @PRT IMSL*DOC.FFTRC) | Access: LIB NBS*IMSL
- FFTRI** Finds the inverse Fourier transform using Fourier coefficients assumed to arise from real data in the time domain. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DFFTRI. | Class(es): J1a1 | Usage: CALL FFTRI (NN,FR,FI) | On-line doc: CALL GAMSDOC FFTRI (or @PRT PORT*DOC.FFTRI) | Access: LIB NBS*PORT | See also: FFTR
- FFTSC** Compute the sine and cosine transforms of a real valued sequence. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): J1a3 | Usage: CALL FFTSC (A,N,ST,CT,IWK,WK,CWK) | On-line doc: CALL GAMSDOC FFTSC (or @PRT IMSL*DOC.FFTSC) | Access: LIB NBS*IMSL
- FIGI** Transforms certain real non-symmetric tridiagonal matrix to symmetric tridiagonal matrix. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1c | Usage: CALL FIGI(NM,N,T,D,E,E2,IERR) | On-line doc: CALL GAMSDOC FIGI (or @PRT CMLIB*DOC.FIGI/EISPACK) | Access: LIB NBS*CMLIB
- FIGI2** Transforms certain real non-symmetric tridiagonal matrix to symmetric tridiagonal matrix. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1c | Usage: CALL FIGI2(NM,N,T,D,E,A,IERR) | On-line doc: CALL GAMSDOC FIGI2 (or @PRT CMLIB*DOC.FIGI2/EISPACK) | Access: LIB NBS*CMLIB | See also: IMTQL2
- FIT** Performs linear least squares regression analysis of a general linear model. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L8a4a1a | Usage: CALL FIT (Y, XM, N, NP, SCRAT, NS, IXM, RES) | On-line doc: CALL GAMSDOC FIT (or @PRT STATLIB*DOC.FIT) | Tests: STATLIB*TEST.DEMO2 | Access: LIB NBS*STATLIB
- FITS** Performs linear least squares regression analysis of a general linear model with computed results returned to the user. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L8a4a1a | Usage: CALL FITS (Y, XM, N, NP, SCRAT, NS, IXM, RES, COEF, PV, SDPV, SDRES, VCV, IVCV, NPRT) | On-line doc: CALL GAMSDOC FITS (or @PRT STATLIB*DOC.FITS) | Tests: STATLIB*TEST.DEMO2 | Access: LIB NBS*STATLIB
- FITW** Performs weighted linear least squares regression analysis of a general linear model. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L8a4a2 | Usage: CALL FITW (Y, XM, WT, N, NP, SCRAT, NS, IXM, RES) | On-line doc: CALL GAMSDOC FITW (or @PRT STATLIB*DOC.FITW) | Tests: STATLIB*TEST.DEMO2 | Access: LIB NBS*STATLIB
- FITWS** Performs weighted linear least squares regression analysis of a general linear model with computed results returned to the user. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L8a4a2 | Usage: CALL FITWS (Y, XM, WT, N, NP, SCRAT, NS, IXM, RES, COEF, PV, SDPV, SDRES, VCV, IVCV, NPRT) | On-line doc: CALL GAMSDOC FITWS (or @PRT STATLIB*DOC.FITWS) | Tests: STATLIB*TEST.DEMO2 | Access: LIB NBS*STATLIB
- FLINV** Inverse Laplace transform of a user supplied complex function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): J3 | Usage: CALL FLINV (F,N,T,ALPHA,NSIG,KMAX,FINV,IER) | On-line doc: CALL GAMSDOC FLINV (or @PRT IMSL*DOC.FLINV) | Access: LIB NBS*IMSL
- FLR** Finds the largest integer less than or equal to x. Input and output are real. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DFLR. | Class(es): C1 | Usage: X = FLR (X) | On-line doc: CALL GAMSDOC FLR (or @PRT PORT*DOC.FLR) | Access: LIB NBS*PORT
- FMIN** Finds an approximate local minimum of a univariate user defined EXTERNAL function, f. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DFMIN. | Class(es): G1a2 | Usage: X = FMIN (F,X,A,B,T) | On-line doc: CALL GAMSDOC FMIN (or @PRT PORT*DOC.FMIN) | Access: LIB NBS*PORT
- FOURIE** Performs a Fourier analysis of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L10a | Usage: CALL FOURIE(X,N) | On-line doc: CALL GAMSDOC FOURIE (or @PRT DATAPAC*DOC.FOURIE) | Access: LIB NBS*DAPAC
- FRAN** Generates a random sample of size N from the F-distribution with degrees of freedom parameters = NU1 and NU2. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a6 | Usage: CALL FRAN(N,NU1,NU2,ISTART,X) | On-line doc: CALL GAMSDOC FRAN (or @PRT DATAPAC*DOC.FRAN) | Access: LIB NBS*DAPAC
- FREQ** Computes the sample frequency and sample cumulative frequency for the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1d | Usage: CALL FREQ(X,N) | On-line doc: CALL GAMSDOC FREQ (or @PRT DATAPAC*DOC.FREQ) | Access: LIB NBS*DAPAC
- FTARPS** Preliminary estimation of the autoregressive parameters in an ARIMA stochastic model. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L10e1 | Usage: CALL FTARPS (ACV,WBAR,IP,IQ,ARPS,PMAC,WA,IER) | On-line doc: CALL GAMSDOC FTARPS (or @PRT IMSL*DOC.FTARPS) | Access: LIB NBS*IMSL | See also: FTAUTO
- FTAUTO** Mean, variance, autocovariances, autocorrelations, and partial autocorrelations for a stationary time series. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L10c | Usage: CALL FTAUTO (W,LW,K,L,ISW,AMEAN,VAR,ACV,AC,PACV,WKAREA) | On-line doc: CALL GAMSDOC FTAUTO (or @PRT IMSL*DOC.FTAUTO) | Access: LIB NBS*IMSL
- FTCAST** Time series forecasts and probability limits using an ARIMA (Box-Jenkins) model. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L10e2 | Usage: CALL FTCAST (Z,ARPS,PMAS,PMAC,ALPHA,LV,DARPS,FCST,WNV,IER) | On-line doc: CALL GAMSDOC FTCAST (or @PRT IMSL*DOC.FTCAST) | Access: LIB NBS*IMSL
- FTCP** Non-seasonal ARIMA (Box-Jenkins) stochastic model analysis for a single time series with full parameter iteration and maximum

likelihood estimation. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L10e | Usage: CALL FTCP (X, IND, DSEED, ALPHA, ARPS, PMAS, PMAC, WNV, FCST, SIM, WK, IER) | On-line doc: CALL GAMSDOC FTCP (or @PRT IMSL*DOC.FTCP) | Access: LIB NBS*IMSL

FTCROS Means, variances, cross-covariances, and cross-correlations for two mutually stationary n channel time series. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L10g1 | Usage: CALL FTCROS (XY, NC, ISW, EMUSIG, ACV, IA, IB, AC, IC, ID, IER) | On-line doc: CALL GAMSDOC FTCROS (or @PRT IMSL*DOC.FTCROS) | Access: LIB NBS*IMSL

FTCRXY Cross-covariance of two mutually stationary time series. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L10g1 | Usage: CALL FTCRXY (X, Y, N, XBAR, YBAR, MLAG, IRS, C, IER) | On-line doc: CALL GAMSDOC FTCRXY (or @PRT IMSL*DOC.FTCRXY) | Access: LIB NBS*IMSL

FTFPS Fast Fourier transform estimates of power spectra and cross spectra of time series. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L10f L10g1 | Usage: CALL FTFPS (X, Y, N, L, IND, PSX, PSY, XPS, IWK, WK, CWK, IER) | On-line doc: CALL GAMSDOC FTFPS (or @PRT IMSL*DOC.FTFPS) | Access: LIB NBS*IMSL

FTFREQ Single or multichannel time series analysis in the time and frequency domains. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L10f L10c1 | Usage: CALL FTFREQ (X, IND, XIND, XYMV, ACV, FREQ, PS, XCOV, XSPECT, AMPHAS, XFER, COHER, IER) | On-line doc: CALL GAMSDOC FTFREQ (or @PRT IMSL*DOC.FTFREQ) | Access: LIB NBS*IMSL

FTGEN Generation of a time series from a given ARIMA (Box-Jenkins) stochastic model. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a20 | Usage: CALL FTGEN (ARPS, PMAS, PMAC, START, WNV, DSEED, IP, IQ, LW, W, WA) | On-line doc: CALL GAMSDOC FTGEN (or @PRT IMSL*DOC.FTGEN) | Access: LIB NBS*IMSL

FTKALM Kalman filtering. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L10b | Usage: CALL FTKALM (K, X, H, G, Y, S, Q, R, P, IN, IS, IL, N, M1, L, T1, T2, IT, T3, IER) | On-line doc: CALL GAMSDOC FTKALM (or @PRT IMSL*DOC.FTKALM) | Access: LIB NBS*IMSL

FTMA Preliminary estimation of the moving average parameters in an ARIMA stochastic model. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L10e1 | Usage: CALL FTMA (ACV, ARPS, IP, IQ, PMAS, WNV, WA, IER) | On-line doc: CALL GAMSDOC FTMA (or @PRT IMSL*DOC.FTMA) | Access: LIB NBS*IMSL | See also: FTAUTO FTARPS

FTML Maximum likelihood estimation of autoregressive and moving average parameters in an ARIMA (Box-Jenkins) stochastic model. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L10e1 | Usage: CALL FTML (X, IND, ARPS, PMAS, PMAC, WNV, GR, A, IER) | On-line doc: CALL GAMSDOC FTML (or @PRT IMSL*DOC.FTML) | Access: LIB NBS*IMSL | See also: FTARPS FTMA

FTRDIF Transformations, differences and seasonal differences of a time series for model identification. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L10a L10b | Usage: CALL FTRDIF (ID1, ID2, IP, IS, LZ, Z, SHIFT, LW, IER) | On-line doc: CALL GAMSDOC FTRDIF (or @PRT IMSL*DOC.FTRDIF) | Access: LIB NBS*IMSL

FTTR Parameter estimates for a univariate transfer function model. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L10g1 | Usage: CALL FTTR (Y, X, N, T, IB, IND, A, IER) | On-line doc: CALL GAMSDOC FTTR (or @PRT IMSL*DOC.FTTR) | Access: LIB NBS*IMSL

FTWEIN Wiener forecast for a stationary stochastic process. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L10e2 | Usage: CALL FTWEIN (WHITE, M, N, EPS, X, LP, WA, IER) | On-line doc: CALL GAMSDOC FTWEIN (or @PRT IMSL*DOC.FTWEIN) | Access: LIB NBS*IMSL

FTWENM Multichannel Wiener forecast. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L10g2 | Usage: CALL FTWENM (AC, NI, M, LF, IA, IB, CC, IC, ID, EPS, TR, IF, IG, F, ERP, LP, WK, IER) | On-line doc: CALL GAMSDOC FTWENM (or @PRT IMSL*DOC.FTWENM) | Access: LIB NBS*IMSL

FTWENX Maximum likelihood parameter estimates for a multichannel, single output time series model. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L10g2 | Usage: CALL FTWENX (X, IX, NS, LS, IP, LAG, ID, R, IR, T, PMAC, WK, IER) | On-line doc: CALL GAMSDOC FTWENX (or @PRT IMSL*DOC.FTWENX) | Access: LIB NBS*IMSL

G

- G01AAE** Simple descriptive statistics, one variable, from raw data. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01AAF. | Class(es): L1a1 | Usage: CALL G01AAE (N, X, IWT, WT, XMEAN, S2, S3, S4, XMIN, XMAX, WTSUM, IFAIL) | On-line doc: CALL GAMSDOC G01AAE (or @PRT NAG*DOC.G01AAE) | Access: LIB NBS*NAG
- G01AAF** Simple descriptive statistics, one variable, from raw data. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01AAE. | Class(es): L1a1 | Usage: CALL G01AAF (N, X, IWT, WT, XMEAN, S2, S3, S4, XMIN, XMAX, WTSUM, IFAIL) | On-line doc: CALL GAMSDOC G01AAF (or @PRT NAG*DOC.G01AAF) | Access: LIB NBS*NAG
- G01ABE** Simple descriptive statistics, two variables, from raw data. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01ABF. | Class(es): L1e1 | Usage: CALL G01ABE (N, X1, X2, IWT, WT, RES, IFAIL) | On-line doc: CALL GAMSDOC G01ABE (or @PRT NAG*DOC.G01ABE) | Access: LIB NBS*NAG
- G01ABF** Simple descriptive statistics, two variables, from raw data. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01ABE. | Class(es): L1e1 | Usage: CALL G01ABF (N, X1, X2, IWT, WT, RES, IFAIL) | On-line doc: CALL GAMSDOC G01ABF (or @PRT NAG*DOC.G01ABF) | Access: LIB NBS*NAG
- G01ADE** Simple descriptive statistics, one variable, from frequency table. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01ADF. | Class(es): L1a3 | Usage: CALL G01ADE (K, X, IFREQ, XMEAN, S2, S3, S4 N, IFAIL) | On-line doc: CALL GAMSDOC G01ADE (or @PRT NAG*DOC.G01ADE) | Access: LIB NBS*NAG
- G01ADF** Simple descriptive statistics, one variable, from frequency table. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01ADE. | Class(es): L1a3 | Usage: CALL G01ADF (K, X, IFREQ, XMEAN, S2, S3, S4 N, IFAIL) | On-line doc: CALL GAMSDOC G01ADF (or @PRT NAG*DOC.G01ADF) | Access: LIB NBS*NAG
- G01AEE** Frequency table from raw data. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01AEF. | Class(es): L2b | Usage: CALL G01AEE (N, K2, X, ICLASS, CINT, IFREQ, XMIN, XMAX, IFAIL) | On-line doc: CALL GAMSDOC G01AEE (or @PRT NAG*DOC.G01AEE) | Access: LIB NBS*NAG
- G01AEF** Frequency table from raw data. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01AEE. | Class(es): L2b | Usage: CALL G01AEF (N, K2, X, ICLASS, CINT, IFREQ, XMIN, XMAX, IFAIL) | On-line doc: CALL GAMSDOC G01AEF (or @PRT NAG*DOC.G01AEF) | Access: LIB NBS*NAG
- G01AFE** Two-way contingency table analysis. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01AFF. | Class(es): L9b | Usage: CALL G01AFE (INOB, IPRED, M, N, NOBS, NUM, PRED, CHIS, P, NPOS, NDF, M1, N1, IFAIL) | On-line doc: CALL GAMSDOC G01AFE (or @PRT NAG*DOC.G01AFE) | Access: LIB NBS*NAG
- G01AFF** Two-way contingency table analysis. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01AFE. | Class(es): L9b | Usage: CALL G01AFF (INOB, IPRED, M, N, NOBS, NUM, PRED, CHIS, P, NPOS, NDF, M1, N1, IFAIL) | On-line doc: CALL GAMSDOC G01AFF (or @PRT NAG*DOC.G01AFF) | Access: LIB NBS*NAG
- G01AGE** Line printer scatter plot of two variables. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01AGF. | Class(es): L3c1 Q1 | Usage: CALL G01AGE (X, Y, NOBS, ISORT, NSTEPX, NSTEPY, IFAIL) | On-line doc: CALL GAMSDOC G01AGE (or @PRT NAG*DOC.G01AGE) | Access: LIB NBS*NAG
- G01AGF** Line printer scatter plot of two variables. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01AGE. | Class(es): L3c1 Q1 | Usage: CALL G01AGF (X, Y, NOBS, ISORT, NSTEPX, NSTEPY, IFAIL) | On-line doc: CALL GAMSDOC G01AGF (or @PRT NAG*DOC.G01AGF) | Access: LIB NBS*NAG
- G01AHE** Line printer scatter plot of one variable against normal scores. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01AHF. | Class(es): L3c4n | Usage: CALL G01AHE (X, NOBS, NSTEPX, NSTEPY, ISTD, IWORK, WORK, LWORK, XSORT, XBAR, XSTD, IFAIL) | On-line doc: CALL GAMSDOC G01AHE (or @PRT NAG*DOC.G01AHE) | Access: LIB NBS*NAG | See also: G01DAF
- G01AHF** Line printer scatter plot of one variable against normal scores. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01AHE. | Class(es): L3c4n | Usage: CALL G01AHF (X, NOBS, NSTEPX, NSTEPY, ISTD, IWORK, WORK, LWORK, XSORT, XBAR, XSTD, IFAIL) | On-line doc: CALL GAMSDOC G01AHF (or @PRT NAG*DOC.G01AHF) | Access: LIB NBS*NAG | See also: G01DAF
- G01AJE** Prints a histogram on a character printing device, with user control over size, positioning, and the range of data values included. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01AJF. | Class(es): L3a | Usage: CALL G01AJE(X,N,NSTEPX,NSTEPY,ITYPE,ISPACE,XMIN,XMAX,XSTEP,N1, MULTY,IFAIL) | On-line doc: CALL GAMSDOC G01AJE (or @PRT NAG*DOC.G01AJE) | Access: LIB NBS*NAG
- G01AJF** Prints a histogram on a character printing device, with user control over size, positioning, and the range of data values included. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01AJE. | Class(es): L3a | Usage: CALL

G01AJF(X,N,NSTEPX,NSTEPY,ITYPE,ISPACE,XMIN,XMAX,XSTEP,N1, MULTY,IFAIL) | On-line doc: CALL GAMSDOC G01AJF
(or @PRT NAG*DOC.G01AJF) | Access: LIB NBS*NAG

G01BAE Student's t distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01BAF. |
Class(es): L5a1t | Usage: D = G01BAE (IDF, T, IFAIL) | On-line doc: CALL GAMSDOC G01BAE (or @PRT NAG*DOC.G01BAE) |
Access: LIB NBS*NAG

G01BAF Student's t distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01BAE. |
Class(es): L5a1t | Usage: D = G01BAF (IDF, T, IFAIL) | On-line doc: CALL GAMSDOC G01BAF (or @PRT NAG*DOC.G01BAF) |
Access: LIB NBS*NAG

G01BBE F (variance ratio) distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version
is G01BBF. | Class(es): L5a1f | Usage: D = G01BBE (I1, I2, A, IFAIL) | On-line doc: CALL GAMSDOC G01BBE (or @PRT
NAG*DOC.G01BBE) | Access: LIB NBS*NAG

G01BBF F (variance ratio) distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version
is G01BBE. | Class(es): L5a1f | Usage: D = G01BBF (I1, I2, A, IFAIL) | On-line doc: CALL GAMSDOC G01BBF (or @PRT
NAG*DOC.G01BBF) | Access: LIB NBS*NAG

G01BCE Chi-square distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01BCF.
| Class(es): L5a1c | Usage: D = G01BCE (X, N, IFAIL) | On-line doc: CALL GAMSDOC G01BCE (or @PRT NAG*DOC.G01BCE) |
Access: LIB NBS*NAG

G01BCF Chi-square distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01BCE.
| Class(es): L5a1c | Usage: D = G01BCF (X, N, IFAIL) | On-line doc: CALL GAMSDOC G01BCF (or @PRT NAG*DOC.G01BCF) |
Access: LIB NBS*NAG

G01BDE Beta distribution of first kind. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01BDF.
| Class(es): L5a1b | Usage: D = G01BDE (X, A, B, IFAIL) | On-line doc: CALL GAMSDOC G01BDE (or @PRT NAG*DOC.G01BDE)
| Access: LIB NBS*NAG

G01BDF Beta distribution of first kind. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01BDE.
| Class(es): L5a1b | Usage: D = G01BDF (X, A, B, IFAIL) | On-line doc: CALL GAMSDOC G01BDF (or @PRT NAG*DOC.G01BDF)
| Access: LIB NBS*NAG

G01CAE Inverse Student's t distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision ver-
sion is G01CAF. | Class(es): L5a2t | Usage: D = G01CAE (P, N, IFAIL) | On-line doc: CALL GAMSDOC G01CAE (or @PRT
NAG*DOC.G01CAE) | Access: LIB NBS*NAG

G01CAF Inverse Student's t distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision ver-
sion is G01CAE. | Class(es): L5a2t | Usage: D = G01CAF (P, N, IFAIL) | On-line doc: CALL GAMSDOC G01CAF (or @PRT
NAG*DOC.G01CAF) | Access: LIB NBS*NAG

G01CBE Inverse F (variance ratio) distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision
version is G01CBF. | Class(es): L5a2f | Usage: D = G01CBE (P, M, N, IFAIL) | On-line doc: CALL GAMSDOC G01CBE (or @PRT
NAG*DOC.G01CBE) | Access: LIB NBS*NAG

G01CBF Inverse F (variance ratio) distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision
version is G01CBE. | Class(es): L5a2f | Usage: D = G01CBF (P, M, N, IFAIL) | On-line doc: CALL GAMSDOC G01CBF (or @PRT
NAG*DOC.G01CBF) | Access: LIB NBS*NAG

G01CCE Inverse chi-square distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version
is G01CCF. | Class(es): L5a2c | Usage: D = G01CCE (P, N, IFAIL) | On-line doc: CALL GAMSDOC G01CCE (or @PRT
NAG*DOC.G01CCE) | Access: LIB NBS*NAG

G01CCF Inverse chi-square distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version
is G01CCE. | Class(es): L5a2c | Usage: D = G01CCF (P, N, IFAIL) | On-line doc: CALL GAMSDOC G01CCF (or @PRT
NAG*DOC.G01CCF) | Access: LIB NBS*NAG

G01CDE Inverse beta distribution of first kind. | Proprietary single precision Fortran subprogram in NAG library. Double precision
version is G01CDF. | Class(es): L5a2b | Usage: D = G01CDE (P, A, B, IFAIL) | On-line doc: CALL GAMSDOC G01CDE (or @PRT
NAG*DOC.G01CDE) | Access: LIB NBS*NAG

G01CDF Inverse beta distribution of first kind. | Proprietary double precision Fortran subprogram in NAG library. Single precision
version is G01CDE. | Class(es): L5a2b | Usage: D = G01CDF (P, A, B, IFAIL) | On-line doc: CALL GAMSDOC G01CDF (or @PRT
NAG*DOC.G01CDF) | Access: LIB NBS*NAG

G01CEE Inverse normal distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01CEF.
| Class(es): L5a2n | Usage: D = G01CEE (P, IFAIL) | On-line doc: CALL GAMSDOC G01CEE (or @PRT NAG*DOC.G01CEE) | Access:
LIB NBS*NAG

G01CEF Inverse normal distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01CEE.

| Class(es): L5a2n | Usage: D = G01CEF (P, IFAIL) | On-line doc: CALL GAMSDOC G01CEF (or @PRT NAG*DOC.G01CEF) | Access: LIB NBS*NAG

G01DAE Calculation of normal scores. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01DAF.
| Class(es): L5a2n | Usage: CALL G01DAE (N, PP, ETOL, ERREST, WORK, IW, IFAIL) | On-line doc: CALL GAMSDOC G01DAE (or @PRT NAG*DOC.G01DAE) | Access: LIB NBS*NAG | See also: G01AHF

G01DAF Calculation of normal scores. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01DAE.
| Class(es): L5a2n | Usage: CALL G01DAF (N, PP, ETOL, ERREST, WORK, IW, IFAIL) | On-line doc: CALL GAMSDOC G01DAF (or @PRT NAG*DOC.G01DAF) | Access: LIB NBS*NAG | See also: G01AHF

G02BAE Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in the array X. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BAF. | Class(es): L1e1 | Usage: CALL G02BAE (N, M, X, IX, XBAR, STD, SSP, ISSP, R, IR, IFAIL) | On-line doc: CALL GAMSDOC G02BAE (or @PRT NAG*DOC.G02BAE) | Access: LIB NBS*NAG

G02BAF Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in the array X. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BAE. | Class(es): L1e1 | Usage: CALL G02BAF (N, M, X, IX, XBAR, STD, SSP, ISSP, R, IR, IFAIL) | On-line doc: CALL GAMSDOC G02BAF (or @PRT NAG*DOC.G02BAF) | Access: LIB NBS*NAG

G02BBE Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in the array X containing missing values. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BBF. | Class(es): L1e2 | Usage: CALL G02BBE (N, M, X, IX, MISS, XMISS, XBAR, STD, SSP, ISSP, R, IR, NCASES, IFAIL) | On-line doc: CALL GAMSDOC G02BBE (or @PRT NAG*DOC.G02BBE) | Access: LIB NBS*NAG

G02BBF Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in the array X containing missing values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BBE. | Class(es): L1e2 | Usage: CALL G02BBF (N, M, X, IX, MISS, XMISS, XBAR, STD, SSP, ISSP, R, IR, NCASES, IFAIL) | On-line doc: CALL GAMSDOC G02BBF (or @PRT NAG*DOC.G02BBF) | Access: LIB NBS*NAG

G02BCE Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in the array X containing missing values. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BCF. | Class(es): L1e2 | Usage: CALL G02BCE (N, M, X, IX, MISS, XMISS, XBAR, STD, SSP, ISSP, R, IR, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BCE (or @PRT NAG*DOC.G02BCE) | Access: LIB NBS*NAG

G02BCF Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in the array X containing missing values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BCE. | Class(es): L1e2 | Usage: CALL G02BCF (N, M, X, IX, MISS, XMISS, XBAR, STD, SSP, ISSP, R, IR, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BCF (or @PRT NAG*DOC.G02BCF) | Access: LIB NBS*NAG

G02BDE Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in the array X. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BDF. | Class(es): L1e1 | Usage: CALL G02BDE (N, M, X, IX, XBAR, STD, SSPZ, ISSPZ, RZ, IRZ, IFAIL) | On-line doc: CALL GAMSDOC G02BDE (or @PRT NAG*DOC.G02BDE) | Access: LIB NBS*NAG

G02BDF Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in the array X. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BDE. | Class(es): L1e1 | Usage: CALL G02BDF (N, M, X, IX, XBAR, STD, SSPZ, ISSPZ, RZ, IRZ, IFAIL) | On-line doc: CALL GAMSDOC G02BDF (or @PRT NAG*DOC.G02BDF) | Access: LIB NBS*NAG

G02BEE Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in array X containing missing values. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BEF. | Class(es): L1e2 | Usage: CALL G02BEE (N, M, X, IX, MISS, XMISS, XBAR, STD, SSPZ, ISSPZ, RZ, IRZ, NCASES, IFAIL) | On-line doc: CALL GAMSDOC G02BEE (or @PRT NAG*DOC.G02BEE) | Access: LIB NBS*NAG

G02BEF Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in array X containing missing values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BEE. | Class(es): L1e2 | Usage: CALL G02BEF (N, M, X, IX, MISS, XMISS, XBAR, STD, SSPZ, ISSPZ, RZ, IRZ, NCASES, IFAIL) | On-line doc: CALL GAMSDOC G02BEF (or @PRT NAG*DOC.G02BEF) | Access: LIB NBS*NAG

G02BFE Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in array X containing missing values. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BFF. | Class(es): L1e2 | Usage: CALL G02BFE (N, M, X, IX, MISS, XMISS, XBAR, STD, SSPZ, ISSPZ, RZ, IRZ, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BFE (or @PRT NAG*DOC.G02BFE) | Access: LIB NBS*NAG

G02BFF Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in array X containing missing values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BFE. | Class(es): L1e2 | Usage: CALL G02BFF (N, M, X, IX, MISS, XMISS, XBAR, STD, SSPZ, ISSPZ, RZ, IRZ, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BFF (or @PRT NAG*DOC.G02BFF) | Access: LIB NBS*NAG

- IRZ, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BFF (or @PRT NAG*DOC.G02BFF) | Access: LIB NBS*NAG
- G02BGE** Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array X. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BGF. | Class(es): L1e1 | Usage: CALL G02BGE (N, M, X, IX, NVARs, KVAR, XBAR, STD, SSP, ISSP, R, IR, IFAIL) | On-line doc: CALL GAMSDOC G02BGE (or @PRT NAG*DOC.G02BGE) | Access: LIB NBS*NAG
- G02BGF** Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array X. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BGE. | Class(es): L1e1 | Usage: CALL G02BGF (N, M, X, IX, NVARs, KVAR, XBAR, STD, SSP, ISSP, R, IR, IFAIL) | On-line doc: CALL GAMSDOC G02BGF (or @PRT NAG*DOC.G02BGF) | Access: LIB NBS*NAG
- G02BHE** Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array X containing missing values. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BHF. | Class(es): L1e2 | Usage: CALL G02BHE (N, M, X, IX, MISS, XMISS, MISTYP, NVARs, KVAR, XBAR, STD, SSP, ISSP, R, IR, NCASES, IFAIL) | On-line doc: CALL GAMSDOC G02BHE (or @PRT NAG*DOC.G02BHE) | Access: LIB NBS*NAG
- G02BHF** Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array X containing missing values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BHE. | Class(es): L1e2 | Usage: CALL G02BHF (N, M, X, IX, MISS, XMISS, MISTYP, NVARs, KVAR, XBAR, STD, SSP, ISSP, R, IR, NCASES, IFAIL) | On-line doc: CALL GAMSDOC G02BHF (or @PRT NAG*DOC.G02BHF) | Access: LIB NBS*NAG
- G02BJE** Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array X containing missing values. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BJF. | Class(es): L1e2 | Usage: CALL G02BJE (N, M, X, IX, MISS, XMISS, NVARs, KVAR, XBAR, STD, SSP, ISSP, R, IR, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BJE (or @PRT NAG*DOC.G02BJE) | Access: LIB NBS*NAG
- G02BJF** Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array X containing missing values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BJE. | Class(es): L1e2 | Usage: CALL G02BJF (N, M, X, IX, MISS, XMISS, NVARs, KVAR, XBAR, STD, SSP, ISSP, R, IR, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BJF (or @PRT NAG*DOC.G02BJF) | Access: LIB NBS*NAG
- G02BKE** Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array X. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BKF. | Class(es): L1e1 | Usage: CALL G02BKE (N, M, X, IX, NVARs, KVAR, XBAR, STD, SSPZ, ISSPZ, RZ, IRZ, IFAIL) | On-line doc: CALL GAMSDOC G02BKE (or @PRT NAG*DOC.G02BKE) | Access: LIB NBS*NAG
- G02BKF** Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array X. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BKE. | Class(es): L1e1 | Usage: CALL G02BKF (N, M, X, IX, NVARs, KVAR, XBAR, STD, SSPZ, ISSPZ, RZ, IRZ, IFAIL) | On-line doc: CALL GAMSDOC G02BKF (or @PRT NAG*DOC.G02BKF) | Access: LIB NBS*NAG
- G02BLE** Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array X containing missing values. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BLF. | Class(es): L1e2 | Usage: CALL G02BLE (N, M, X, IX, MISS, XMISS, MISTYP, NVARs, KVAR, XBAR, STD, SSPZ, ISSPZ, RZ, IRZ, NCASES, IFAIL) | On-line doc: CALL GAMSDOC G02BLE (or @PRT NAG*DOC.G02BLE) | Access: LIB NBS*NAG
- G02BLF** Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array X containing missing values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BLE. | Class(es): L1e2 | Usage: CALL G02BLF (N, M, X, IX, MISS, XMISS, MISTYP, NVARs, KVAR, XBAR, STD, SSPZ, ISSPZ, RZ, IRZ, NCASES, IFAIL) | On-line doc: CALL GAMSDOC G02BLF (or @PRT NAG*DOC.G02BLF) | Access: LIB NBS*NAG
- G02BME** Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array X containing missing values. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BMF. | Class(es): L1e2 | Usage: CALL G02BME (N, M, X, IX, MISS, XMISS, NVARs, KVAR, XBAR, STD, SSPZ, ISSPZ, RZ, IRZ, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BME (or @PRT NAG*DOC.G02BME) | Access: LIB NBS*NAG
- G02BMF** Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array X containing missing values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BME. | Class(es): L1e2 | Usage: CALL G02BMF (N, M, X, IX, MISS, XMISS, NVARs, KVAR, XBAR, STD, SSPZ, ISSPZ, RZ, IRZ, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BMF (or @PRT NAG*DOC.G02BMF) | Access: LIB NBS*NAG

- G02BNE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X, overwriting X with the ranks of the observations. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BNF. | Class(es): L1e1b | Usage: CALL G02BNE (N, M, X, IX, ITYPE, RR, IRR, KWORKA, KWORKB, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BNE (or @PRT NAG*DOC.G02BNE) | Access: LIB NBS*NAG
- G02BNF** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X, overwriting X with the ranks of the observations. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BNE. | Class(es): L1e1b | Usage: CALL G02BNF (N, M, X, IX, ITYPE, RR, IRR, KWORKA, KWORKB, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BNF (or @PRT NAG*DOC.G02BNF) | Access: LIB NBS*NAG
- G02BPE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X containing missing values, overwriting X with the ranks of the observations. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BPF. | Class(es): L1e1b | Usage: CALL G02BPE (N, M, X, IX, MISS, XMISS, ITYPE, RR, IRR, NCASES, INCASE, KWORKA, KWORKB, KWORKC, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BPE (or @PRT NAG*DOC.G02BPE) | Access: LIB NBS*NAG
- G02BPF** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X containing missing values, overwriting X with the ranks of the observations. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BPE. | Class(es): L1e1b | Usage: CALL G02BPF (N, M, X, IX, MISS, XMISS, ITYPE, RR, IRR, NCASES, INCASE, KWORKA, KWORKB, KWORKC, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BPF (or @PRT NAG*DOC.G02BPF) | Access: LIB NBS*NAG
- G02BQE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X, preserving X. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BQF. | Class(es): L1e1b | Usage: CALL G02BQE (N, M, X, IX, ITYPE, RR, IRR, KWORKA, KWORKB, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BQE (or @PRT NAG*DOC.G02BQE) | Access: LIB NBS*NAG
- G02BQF** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X, preserving X. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BQE. | Class(es): L1e1b | Usage: CALL G02BQF (N, M, X, IX, ITYPE, RR, IRR, KWORKA, KWORKB, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BQF (or @PRT NAG*DOC.G02BQF) | Access: LIB NBS*NAG
- G02BRE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X containing missing values, preserving X. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BRF. | Class(es): L1e1b | Usage: CALL G02BRE (N, M, X, IX, MISS, XMISS, ITYPE, RR, IRR, NCASES, INCASE, KWORKA, KWORKB, KWORKC, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BRE (or @PRT NAG*DOC.G02BRE) | Access: LIB NBS*NAG
- G02BRF** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X containing missing values, preserving X. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BRE. | Class(es): L1e1b | Usage: CALL G02BRF (N, M, X, IX, MISS, XMISS, ITYPE, RR, IRR, NCASES, INCASE, KWORKA, KWORKB, KWORKC, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BRF (or @PRT NAG*DOC.G02BRF) | Access: LIB NBS*NAG
- G02BSE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X containing missing values, preserving X. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BSF. | Class(es): L1e1b | Usage: CALL G02BSE (N, M, X, IX, MISS, XMISS, ITYPE, RR, IRR, NCASES, COUNT, IC, KWORKA, KWORKB, KWORKC, KWORKD, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BSE (or @PRT NAG*DOC.G02BSE) | Access: LIB NBS*NAG
- G02BSF** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X containing missing values, preserving X. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BSE. | Class(es): L1e1b | Usage: CALL G02BSF (N, M, X, IX, MISS, XMISS, ITYPE, RR, IRR, NCASES, COUNT, IC, KWORKA, KWORKB, KWORKC, KWORKD, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BSF (or @PRT NAG*DOC.G02BSF) | Access: LIB NBS*NAG
- G02CAE** Simple linear regression with constant term, no missing values. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02CAF. | Class(es): L8a1a1a | Usage: CALL G02CAE (N, X, Y, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CAE (or @PRT NAG*DOC.G02CAE) | Access: LIB NBS*NAG
- G02CAF** Simple linear regression with constant term, no missing values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02CAE. | Class(es): L8a1a1a | Usage: CALL G02CAF (N, X, Y, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CAF (or @PRT NAG*DOC.G02CAF) | Access: LIB NBS*NAG
- G02CBE** Simple linear regression without constant term, no missing values. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02CBF. | Class(es): L8a1b | Usage: CALL G02CBE (N, X, Y, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CBE (or @PRT NAG*DOC.G02CBE) | Access: LIB NBS*NAG
- G02CBF** Simple linear regression without constant term, no missing values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02CBE. | Class(es): L8a1b | Usage: CALL G02CBF (N, X, Y, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CBF (or @PRT NAG*DOC.G02CBF) | Access: LIB NBS*NAG
- G02CCE** Simple linear regression with constant term, missing values. | Proprietary single precision Fortran subprogram in NAG library.

Double precision version is G02CCF. | Class(es): L8a1a1b | Usage: CALL G02CCE (N, X, Y, XMISS, YMISS, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CCE (or @PRT NAG*DOC.G02CCE) | Access: LIB NBS*NAG

G02CCF Simple linear regression with constant term, missing values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02CCE. | Class(es): L8a1a1b | Usage: CALL G02CCF (N, X, Y, XMISS, YMISS, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CCF (or @PRT NAG*DOC.G02CCF) | Access: LIB NBS*NAG

G02CDE Simple linear regression without constant term, missing values. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02CDF. | Class(es): L8a1b | Usage: CALL G02CDE (N, X, Y, XMISS, YMISS, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CDE (or @PRT NAG*DOC.G02CDE) | Access: LIB NBS*NAG

G02CDF Simple linear regression without constant term, missing values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02CDE. | Class(es): L8a1b | Usage: CALL G02CDF (N, X, Y, XMISS, YMISS, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CDF (or @PRT NAG*DOC.G02CDF) | Access: LIB NBS*NAG

G02CEE Service routines for multiple linear regression, select elements from vectors and matrices. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02CEF. | Class(es): L8h L2d | Usage: CALL G02CEE (NVAR, XBAR, STD, SSP, ISSP, R, IR, NVARS2, KORDER, XBAR2, STD2, SSP2, ISSP2, R2, IR2, IFAIL) | On-line doc: CALL GAMSDOC G02CEE (or @PRT NAG*DOC.G02CEE) | Access: LIB NBS*NAG

G02CEF Service routines for multiple linear regression, select elements from vectors and matrices. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02CEE. | Class(es): L8h L2d | Usage: CALL G02CEF (NVAR, XBAR, STD, SSP, ISSP, R, IR, NVARS2, KORDER, XBAR2, STD2, SSP2, ISSP2, R2, IR2, IFAIL) | On-line doc: CALL GAMSDOC G02CEF (or @PRT NAG*DOC.G02CEF) | Access: LIB NBS*NAG

G02CFE Service routines for multiple linear regression, re-order elements of vectors and matrices. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02CFF. | Class(es): L8h | Usage: CALL G02CFE (NVAR, KORDER, XBAR, STD, SSP, ISSP, R, IR, KWORK, IFAIL) | On-line doc: CALL GAMSDOC G02CFE (or @PRT NAG*DOC.G02CFE) | Access: LIB NBS*NAG

G02CFF Service routines for multiple linear regression, re-order elements of vectors and matrices. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02CFE. | Class(es): L8h | Usage: CALL G02CFF (NVAR, KORDER, XBAR, STD, SSP, ISSP, R, IR, KWORK, IFAIL) | On-line doc: CALL GAMSDOC G02CFF (or @PRT NAG*DOC.G02CFF) | Access: LIB NBS*NAG

G02CGE Performs a multiple linear regression on the set of variables whose means, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients are given. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02CGF. | Class(es): L8a4a1c | Usage: CALL G02CGE (NCASES, NVARS, NIND, XBAR, SSP, ISSP, R, IR, RESULT, COEFF, ICOEFF, CONST, RINV, IRINV, C, IC, WKZ, IWKZ, IFAIL) | On-line doc: CALL GAMSDOC G02CGE (or @PRT NAG*DOC.G02CGE) | Access: LIB NBS*NAG

G02CGF Performs a multiple linear regression on the set of variables whose means, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients are given. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02CGE. | Class(es): L8a4a1c | Usage: CALL G02CGF (NCASES, NVARS, NIND, XBAR, SSP, ISSP, R, IR, RESULT, COEFF, ICOEFF, CONST, RINV, IRINV, C, IC, WKZ, IWKZ, IFAIL) | On-line doc: CALL GAMSDOC G02CGF (or @PRT NAG*DOC.G02CGF) | Access: LIB NBS*NAG

G02CHE Performs a multiple linear regression with no constant on the set of variables whose sums of squares and cross-products about zero and correlation-like coefficients are given. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02CHF. | Class(es): L8a4a1c | Usage: CALL G02CHE (NCASES, NVARS, NIND, SSPZ, ISSPZ, RZ, IRZ, RESULT, COEFF, ICOEFF, RZINV, IRZINV, CZ, ICZ, WKZ, IWKZ, IFAIL) | On-line doc: CALL GAMSDOC G02CHE (or @PRT NAG*DOC.G02CHE) | Access: LIB NBS*NAG

G02CHF Performs a multiple linear regression with no constant on the set of variables whose sums of squares and cross-products about zero and correlation-like coefficients are given. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02CHE. | Class(es): L8a4a1c | Usage: CALL G02CHF (NCASES, NVARS, NIND, SSPZ, ISSPZ, RZ, IRZ, RESULT, COEFF, ICOEFF, RZINV, IRZINV, CZ, ICZ, WKZ, IWKZ, IFAIL) | On-line doc: CALL GAMSDOC G02CHF (or @PRT NAG*DOC.G02CHF) | Access: LIB NBS*NAG

G02CJE Performs one or more multiple linear regressions, regressing each of a set of dependent variables separately on the same set of independent variables. Input is raw data; output includes, for each dependent variable, estimates of regression coefficients and an estimate of the variance of residuals. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02CJF. | Class(es): L8a7 | Usage: CALL G02CJE (X, IX, Y, IY, N, M, IR, THETA, IT, SIGSQ, C, IC, IPIV, WK1, WK2, IFAIL) | On-line doc: CALL GAMSDOC G02CJE (or @PRT NAG*DOC.G02CJE) | Access: LIB NBS*NAG

G02CJF Performs one or more multiple linear regressions, regressing each of a set of dependent variables separately on the same set of independent variables. Input is raw data; output includes, for each dependent variable, estimates of regression coefficients and an estimate of the variance of residuals. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02CJE. | Class(es): L8a7 | Usage: CALL G02CJF (X, IX, Y, IY, N, M, IR, THETA, IT, SIGSQ, C, IC, IPIV, WK1, WK2, IFAIL) | On-line doc: CALL GAMSDOC G02CJF (or @PRT NAG*DOC.G02CJF) | Access: LIB NBS*NAG

- G04ADE** Three-way analysis of variance, Latin square design. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G04ADF. | Class(es): L7a2b1a | Usage: CALL G04ADE (DATA, VAR, AMR, AMC, AMT, LCODE, IA, N, NN) | On-line doc: CALL GAMSDOC G04ADE (or @PRT NAG*DOC.G04ADE) | Access: LIB NBS*NAG
- G04ADF** Three-way analysis of variance, Latin square design. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G04ADE. | Class(es): L7a2b1a | Usage: CALL G04ADF (DATA, VAR, AMR, AMC, AMT, LCODE, IA, N, NN) | On-line doc: CALL GAMSDOC G04ADF (or @PRT NAG*DOC.G04ADF) | Access: LIB NBS*NAG
- G04AEE** One-way analysis of variance, subgroups of unequal size. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G04AEF. | Class(es): L7a1a | Usage: CALL G04AEE (Y, N, K, NOBS, GBAR, GM, SS, IDF, F, FP, IFAIL) | On-line doc: CALL GAMSDOC G04AEE (or @PRT NAG*DOC.G04AEE) | Access: LIB NBS*NAG
- G04AEF** One-way analysis of variance, subgroups of unequal size. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G04AEE. | Class(es): L7a1a | Usage: CALL G04AEF (Y, N, K, NOBS, GBAR, GM, SS, IDF, F, FP, IFAIL) | On-line doc: CALL GAMSDOC G04AEF (or @PRT NAG*DOC.G04AEF) | Access: LIB NBS*NAG
- G04AFE** Two-way analysis of variance, cross-classification, subgroups of equal size. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G04AFF. | Class(es): L7a2a1a | Usage: CALL G04AFE (Y, IY1, IY2, M, NR, NC, ROW, COL, CELL, ICELL, GM, SS, IDF, F, FP, IFAIL) | On-line doc: CALL GAMSDOC G04AFE (or @PRT NAG*DOC.G04AFE) | Access: LIB NBS*NAG
- G04AFF** Two-way analysis of variance, cross-classification, subgroups of equal size. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G04AFE. | Class(es): L7a2a1a | Usage: CALL G04AFF (Y, IY1, IY2, M, NR, NC, ROW, COL, CELL, ICELL, GM, SS, IDF, F, FP, IFAIL) | On-line doc: CALL GAMSDOC G04AFF (or @PRT NAG*DOC.G04AFF) | Access: LIB NBS*NAG
- G04AGE** Two-way analysis of variance, hierarchial classification, subgroups of unequal size. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G04AGF. | Class(es): L7a4a | Usage: CALL G04AGE (Y, N, K, LSUB, NOBS, L, NGP, GBAR, SGBAR, GM, SS, IDF, F, FP, IFAIL) | On-line doc: CALL GAMSDOC G04AGE (or @PRT NAG*DOC.G04AGE) | Access: LIB NBS*NAG
- G04AGF** Two-way analysis of variance, hierarchial classification, subgroups of unequal size. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G04AGE. | Class(es): L7a4a | Usage: CALL G04AGF (Y, N, K, LSUB, NOBS, L, NGP, GBAR, SGBAR, GM, SS, IDF, F, FP, IFAIL) | On-line doc: CALL GAMSDOC G04AGF (or @PRT NAG*DOC.G04AGF) | Access: LIB NBS*NAG
- G05CAE** Pseudo-random real numbers, uniform distribution over (0.0,1.0). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05CAF. | Class(es): L6a21 | Usage: D = G05CAE (X) | On-line doc: CALL GAMSDOC G05CAE (or @PRT NAG*DOC.G05CAE) | Access: LIB NBS*NAG
- G05CAF** Pseudo-random real numbers, uniform distribution over (0.0,1.0). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05CAE. | Class(es): L6a21 | Usage: D = G05CAF (X) | On-line doc: CALL GAMSDOC G05CAF (or @PRT NAG*DOC.G05CAF) | Access: LIB NBS*NAG
- G05CBE** Initialise random number generating routines, to give a repeatable sequence. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05CBF. | Class(es): L6c | Usage: CALL G05CBE (I) | On-line doc: CALL GAMSDOC G05CBE (or @PRT NAG*DOC.G05CBE) | Access: LIB NBS*NAG
- G05CBF** Initialise random number generating routines, to give a repeatable sequence. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05CBE. | Class(es): L6c | Usage: CALL G05CBF (I) | On-line doc: CALL GAMSDOC G05CBF (or @PRT NAG*DOC.G05CBF) | Access: LIB NBS*NAG
- G05CCE** Initialise random number generating routines, to give nonrepeatable sequence. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05CCF. | Class(es): L6c | Usage: CALL G05CCE | On-line doc: CALL GAMSDOC G05CCE (or @PRT NAG*DOC.G05CCE) | Access: LIB NBS*NAG
- G05CCF** Initialise random number generating routines, to give nonrepeatable sequence. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05CCE. | Class(es): L6c | Usage: CALL G05CCF | On-line doc: CALL GAMSDOC G05CCF (or @PRT NAG*DOC.G05CCF) | Access: LIB NBS*NAG
- G05CFE** Save state of random number generating routines. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05CFF. | Class(es): L6c | Usage: CALL G05CFE (IA, NI, XA, NX, IFAIL) | On-line doc: CALL GAMSDOC G05CFE (or @PRT NAG*DOC.G05CFE) | Access: LIB NBS*NAG
- G05CFF** Save state of random number generating routines. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05CFE. | Class(es): L6c | Usage: CALL G05CFF (IA, NI, XA, NX, IFAIL) | On-line doc: CALL GAMSDOC G05CFF (or @PRT NAG*DOC.G05CFF) | Access: LIB NBS*NAG
- G05CGE** Restore state of random number generating routines. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05CGF. | Class(es): L6c | Usage: CALL G05CGE (IA, NI, XA, NX, IFAIL) | On-line doc: CALL GAMSDOC

G05CGE (or @PRT NAG*DOC.G05CGE) | Access: LIB NBS*NAG

G05CGF Restore state of random number generating routines. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05CGE. | Class(es): L6c | Usage: CALL G05CGF (IA, NI, XA, NX, IFAIL) | On-line doc: CALL GAMSDOC G05CGF (or @PRT NAG*DOC.G05CGF) | Access: LIB NBS*NAG

G05DAE Pseudo-random real numbers, uniform distribution over (a,b). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DAF. | Class(es): L6a21 | Usage: D = G05DAE (A, B) | On-line doc: CALL GAMSDOC G05DAE (or @PRT NAG*DOC.G05DAE) | Access: LIB NBS*NAG

G05DAF Pseudo-random real numbers, uniform distribution over (a,b). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DAE. | Class(es): L6a21 | Usage: D = G05DAF (A, B) | On-line doc: CALL GAMSDOC G05DAF (or @PRT NAG*DOC.G05DAF) | Access: LIB NBS*NAG

G05DBE Pseudo-random real numbers, exponential distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DBF. | Class(es): L6a5 | Usage: D = G05DBE (A) | On-line doc: CALL GAMSDOC G05DBE (or @PRT NAG*DOC.G05DBE) | Access: LIB NBS*NAG

G05DBF Pseudo-random real numbers, exponential distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DBE. | Class(es): L6a5 | Usage: D = G05DBF (A) | On-line doc: CALL GAMSDOC G05DBF (or @PRT NAG*DOC.G05DBF) | Access: LIB NBS*NAG

G05DCE Pseudo-random real numbers, logistic distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DCF. | Class(es): L6a12 | Usage: D = G05DCE (A, B) | On-line doc: CALL GAMSDOC G05DCE (or @PRT NAG*DOC.G05DCE) | Access: LIB NBS*NAG

G05DCF Pseudo-random real numbers, logistic distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DCE. | Class(es): L6a12 | Usage: D = G05DCF (A, B) | On-line doc: CALL GAMSDOC G05DCF (or @PRT NAG*DOC.G05DCF) | Access: LIB NBS*NAG

G05DDE Pseudo-random real numbers, normal distribution (a,b). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DDF. | Class(es): L6a14 | Usage: D = G05DDE (A, B) | On-line doc: CALL GAMSDOC G05DDE (or @PRT NAG*DOC.G05DDE) | Access: LIB NBS*NAG

G05DDF Pseudo-random real numbers, normal distribution (a,b). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DDE. | Class(es): L6a14 | Usage: D = G05DDF (A, B) | On-line doc: CALL GAMSDOC G05DDF (or @PRT NAG*DOC.G05DDF) | Access: LIB NBS*NAG

G05DEE Pseudo-random real numbers, lognormal distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DEF. | Class(es): L6a12 | Usage: D = G05DEE (A, B) | On-line doc: CALL GAMSDOC G05DEE (or @PRT NAG*DOC.G05DEE) | Access: LIB NBS*NAG

G05DEF Pseudo-random real numbers, lognormal distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DEE. | Class(es): L6a12 | Usage: D = G05DEF (A, B) | On-line doc: CALL GAMSDOC G05DEF (or @PRT NAG*DOC.G05DEF) | Access: LIB NBS*NAG

G05DFE Pseudo-random real numbers, Cauchy distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DFF. | Class(es): L6a3 | Usage: D = G05DFE (A, B) | On-line doc: CALL GAMSDOC G05DFE (or @PRT NAG*DOC.G05DFE) | Access: LIB NBS*NAG

G05DFF Pseudo-random real numbers, Cauchy distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DFE. | Class(es): L6a3 | Usage: D = G05DFF (A, B) | On-line doc: CALL GAMSDOC G05DFF (or @PRT NAG*DOC.G05DFF) | Access: LIB NBS*NAG

G05DGE Pseudo-random real numbers, gamma distribution with parameters (g,h). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DGF. | Class(es): L6a7 | Usage: D = G05DGE (G, H, IFAIL) | On-line doc: CALL GAMSDOC G05DGE (or @PRT NAG*DOC.G05DGE) | Access: LIB NBS*NAG

G05DGF Pseudo-random real numbers, gamma distribution with parameters (g,h). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DGE. | Class(es): L6a7 | Usage: D = G05DGF (G, H, IFAIL) | On-line doc: CALL GAMSDOC G05DGF (or @PRT NAG*DOC.G05DGF) | Access: LIB NBS*NAG

G05DHE Pseudo-random real numbers, chi-square distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DHF. | Class(es): L6a3 | Usage: D = G05DHE (N, IFAIL) | On-line doc: CALL GAMSDOC G05DHE (or @PRT NAG*DOC.G05DHE) | Access: LIB NBS*NAG

G05DHF Pseudo-random real numbers, chi-square distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DHE. | Class(es): L6a3 | Usage: D = G05DHF (N, IFAIL) | On-line doc: CALL GAMSDOC G05DHF (or @PRT

NAG*DOC.G05DHF) | Access: LIB NBS*NAG

G05DJE Pseudo-random real numbers, Student's t distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DJF. | Class(es): L6a20 | Usage: D = G05DJE (N, IFAIL) | On-line doc: CALL GAMSDOC G05DJE (or @PRT NAG*DOC.G05DJE) | Access: LIB NBS*NAG

G05DJF Pseudo-random real numbers, Student's t distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DJE. | Class(es): L6a20 | Usage: D = G05DJF (N, IFAIL) | On-line doc: CALL GAMSDOC G05DJF (or @PRT NAG*DOC.G05DJF) | Access: LIB NBS*NAG

G05DKE Pseudo-random real numbers, Snedecor's F-distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DKF. | Class(es): L6a8 | Usage: D = G05DKE (M, N, IFAIL) | On-line doc: CALL GAMSDOC G05DKE (or @PRT NAG*DOC.G05DKE) | Access: LIB NBS*NAG

G05DKF Pseudo-random real numbers, Snedecor's F-distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DKE. | Class(es): L6a8 | Usage: D = G05DKF (M, N, IFAIL) | On-line doc: CALL GAMSDOC G05DKF (or @PRT NAG*DOC.G05DKF) | Access: LIB NBS*NAG

G05DLE Pseudo-random real numbers, Beta distribution of the first kind. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DLF. | Class(es): L6a2 | Usage: D = G05DLE (G, H, IFAIL) | On-line doc: CALL GAMSDOC G05DLE (or @PRT NAG*DOC.G05DLE) | Access: LIB NBS*NAG

G05DLF Pseudo-random real numbers, Beta distribution of the first kind. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DLE. | Class(es): L6a2 | Usage: D = G05DLF (G, H, IFAIL) | On-line doc: CALL GAMSDOC G05DLF (or @PRT NAG*DOC.G05DLF) | Access: LIB NBS*NAG

G05DME Pseudo-random real numbers, Beta distribution of the second kind. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DMF. | Class(es): L6a2 | Usage: D = G05DME (G, H, IFAIL) | On-line doc: CALL GAMSDOC G05DME (or @PRT NAG*DOC.G05DME) | Access: LIB NBS*NAG

G05DMF Pseudo-random real numbers, Beta distribution of the second kind. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DME. | Class(es): L6a2 | Usage: D = G05DMF (G, H, IFAIL) | On-line doc: CALL GAMSDOC G05DMF (or @PRT NAG*DOC.G05DMF) | Access: LIB NBS*NAG

G05DPE Pseudo-random real numbers, Weibull distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DPF. | Class(es): L6a23 | Usage: D = G05DPE (A, B, IFAIL) | On-line doc: CALL GAMSDOC G05DPE (or @PRT NAG*DOC.G05DPE) | Access: LIB NBS*NAG

G05DPF Pseudo-random real numbers, Weibull distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DPE. | Class(es): L6a23 | Usage: D = G05DPF (A, B, IFAIL) | On-line doc: CALL GAMSDOC G05DPF (or @PRT NAG*DOC.G05DPF) | Access: LIB NBS*NAG

G05DYE Pseudo-random integer from uniform distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DYF. | Class(es): L6a21 L6a9 | Usage: I = G05DYE (M, N) | On-line doc: CALL GAMSDOC G05DYE (or @PRT NAG*DOC.G05DYE) | Access: LIB NBS*NAG

G05DYF Pseudo-random integer from uniform distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DYE. | Class(es): L6a21 L6a9 | Usage: I = G05DYF (M, N) | On-line doc: CALL GAMSDOC G05DYF (or @PRT NAG*DOC.G05DYF) | Access: LIB NBS*NAG

G05DZE Pseudo-random logical value. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DZF. | Class(es): L6a2 L6a12 | Usage: L = G05DZE (P) | On-line doc: CALL GAMSDOC G05DZE (or @PRT NAG*DOC.G05DZE) | Access: LIB NBS*NAG

G05DZF Pseudo-random logical value. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DZE. | Class(es): L6a2 L6a12 | Usage: L = G05DZF (P) | On-line doc: CALL GAMSDOC G05DZF (or @PRT NAG*DOC.G05DZF) | Access: LIB NBS*NAG

G05EBE Set up reference vector for generating pseudo-random integers, uniform distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05EBF. | Class(es): L6a21 L6a9 | Usage: CALL G05EBE (M, N, R, NR, IFAIL) | On-line doc: CALL GAMSDOC G05EBE (or @PRT NAG*DOC.G05EBE) | Access: LIB NBS*NAG | See also: G05EYE

G05EBF Set up reference vector for generating pseudo-random integers, uniform distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05EBE. | Class(es): L6a21 L6a9 | Usage: CALL G05EBF (M, N, R, NR, IFAIL) | On-line doc: CALL GAMSDOC G05EBF (or @PRT NAG*DOC.G05EBF) | Access: LIB NBS*NAG | See also: G05EYF

G05ECE Set up reference vector for generating pseudo-random integers, Poisson distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05ECF. | Class(es): L6a16 | Usage: CALL G05ECE (T, R, NR, IFAIL) | On-line doc: CALL GAMSDOC G05ECE (or @PRT NAG*DOC.G05ECE) | Access: LIB NBS*NAG | See also: G05EYE

- G05ECF** Set up reference vector for generating pseudo-random integers, Poisson distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05ECE. | Class(es): L6a16 | Usage: CALL G05ECF (T, R, NR, IFAIL) | On-line doc: CALL GAMSDOC G05ECF (or @PRT NAG*DOC.G05ECF) | Access: LIB NBS*NAG | See also: G05EYF
- G05EDE** Set up reference vector for generating pseudo-random integers, binomial distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05EDF. | Class(es): L6a2 | Usage: CALL G05EDE (N, P, R, NR, IFAIL) | On-line doc: CALL GAMSDOC G05EDE (or @PRT NAG*DOC.G05EDE) | Access: LIB NBS*NAG | See also: G05EYE
- G05EDF** Set up reference vector for generating pseudo-random integers, binomial distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05EDE. | Class(es): L6a2 | Usage: CALL G05EDF (N, P, R, NR, IFAIL) | On-line doc: CALL GAMSDOC G05EDF (or @PRT NAG*DOC.G05EDF) | Access: LIB NBS*NAG | See also: G05EYF
- G05EEE** Set up reference vector for generating pseudo-random integers, negative binomial distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05EEF. | Class(es): L6a14 | Usage: CALL G05EEE (N, P, R, NR, IFAIL) | On-line doc: CALL GAMSDOC G05EEE (or @PRT NAG*DOC.G05EEE) | Access: LIB NBS*NAG | See also: G05EYE
- G05EEF** Set up reference vector for generating pseudo-random integers, negative binomial distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05EEE. | Class(es): L6a14 | Usage: CALL G05EEF (N, P, R, NR, IFAIL) | On-line doc: CALL GAMSDOC G05EEF (or @PRT NAG*DOC.G05EEF) | Access: LIB NBS*NAG | See also: G05EYF
- G05EFE** Set up reference vector for generating pseudo-random integers, hypergeometric distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05EFF. | Class(es): L6a8 | Usage: CALL G05EFE (L, M, N, R, NR, IFAIL) | On-line doc: CALL GAMSDOC G05EFE (or @PRT NAG*DOC.G05EFE) | Access: LIB NBS*NAG | See also: G05EYE
- G05EFF** Set up reference vector for generating pseudo-random integers, hypergeometric distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05EFE. | Class(es): L6a8 | Usage: CALL G05EFF (L, M, N, R, NR, IFAIL) | On-line doc: CALL GAMSDOC G05EFF (or @PRT NAG*DOC.G05EFF) | Access: LIB NBS*NAG | See also: G05EYF
- G05EGE** Set up reference vector for univariate ARMA time series model. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05EGF. | Class(es): L6a20 | Usage: CALL G05EGE (E, A, NA, B, NB, R, NR, VAR, IFAIL) | On-line doc: CALL GAMSDOC G05EGE (or @PRT NAG*DOC.G05EGE) | Access: LIB NBS*NAG | See also: G05EWE
- G05EGF** Set up reference vector for univariate ARMA time series model. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05EGE. | Class(es): L6a20 | Usage: CALL G05EGF (E, A, NA, B, NB, R, NR, VAR, IFAIL) | On-line doc: CALL GAMSDOC G05EGF (or @PRT NAG*DOC.G05EGF) | Access: LIB NBS*NAG | See also: G05EWF
- G05EHE** Performs a pseudo-random permutation of a vector of integers. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05EHF. | Class(es): L6a16 | Usage: CALL G05EHE(INDEX,N,IFAIL) | On-line doc: CALL GAMSDOC G05EHE (or @PRT NAG*DOC.G05EHE) | Access: LIB NBS*NAG
- G05EHF** Performs a pseudo-random permutation of a vector of integers. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05EHE. | Class(es): L6a16 | Usage: CALL G05EHF(INDEX,N,IFAIL) | On-line doc: CALL GAMSDOC G05EHF (or @PRT NAG*DOC.G05EHF) | Access: LIB NBS*NAG
- G05EJE** Selects a pseudo-random sample from an integer vector. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05EJF. | Class(es): L6a19 | Usage: CALL G05EJE(IA,N,IB,M,IFAIL) | On-line doc: CALL GAMSDOC G05EJE (or @PRT NAG*DOC.G05EJE) | Access: LIB NBS*NAG
- G05EJF** Selects a pseudo-random sample from an integer vector. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05EJE. | Class(es): L6a19 | Usage: CALL G05EJF(IA,N,IB,M,IFAIL) | On-line doc: CALL GAMSDOC G05EJF (or @PRT NAG*DOC.G05EJF) | Access: LIB NBS*NAG
- G05EWE** Generate next term from ARMA time series using vector from G05EGE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05EWF. | Class(es): L6a20 | Usage: D = G05EWE (R, NR, IFAIL) | On-line doc: CALL GAMSDOC G05EWE (or @PRT NAG*DOC.G05EWE) | Access: LIB NBS*NAG | See also: G05EGE
- G05EWF** Generate next term from ARMA time series using vector from G05EGF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05EWE. | Class(es): L6a20 | Usage: D = G05EWF (R, NR, IFAIL) | On-line doc: CALL GAMSDOC G05EWF (or @PRT NAG*DOC.G05EWF) | Access: LIB NBS*NAG | See also: G05EGF
- G05EXE** Set up reference vector from supplied cumulative distribution function or probability distribution function. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05EXF. | Class(es): L6a7 | Usage: CALL G05EXE (P, NP, IP, LP, R, NR, IFAIL) | On-line doc: CALL GAMSDOC G05EXE (or @PRT NAG*DOC.G05EXE) | Access: LIB NBS*NAG | See also: G05EYE
- G05EXF** Set up reference vector from supplied cumulative distribution function or probability distribution function. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05EXE. | Class(es): L6a7 | Usage: CALL G05EXF (P, NP, IP, LP, R, NR, IFAIL) | On-line doc: CALL GAMSDOC G05EXF (or @PRT NAG*DOC.G05EXF) | Access: LIB NBS*NAG | See also: G05EYF

- G05EYE** Pseudo-random integer from reference vector. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05EYF. | Class(es): L6a9 | Usage: 1 = G05EYE (R, NR) | On-line doc: CALL GAMSDOC G05EYE (or @PRT NAG*DOC.G05EYE) | Access: LIB NBS*NAG | See also: G05EBE G05ECE G05EDE G05EEE G05EFE G05EXE
- G05EYF** Pseudo-random integer from reference vector. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05EYE. | Class(es): L6a9 | Usage: 1 = G05EYF (R, NR) | On-line doc: CALL GAMSDOC G05EYF (or @PRT NAG*DOC.G05EYF) | Access: LIB NBS*NAG | See also: G05EBF G05ECF G05EDF G05EEF G05EFF G05EXF
- G05EZE** Returns a pseudo-random multivariate normal vector taken from a distribution described by a reference vector set up by G05EAE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05EZF. | Class(es): L6b14 | Usage: CALL G05EZE(Z,N,R,NR,IFAIL) | On-line doc: CALL GAMSDOC G05EZE (or @PRT NAG*DOC.G05EZE) | Access: LIB NBS*NAG | See also: G05EAE
- G05EZF** Returns a pseudo-random multivariate normal vector taken from a distribution described by a reference vector set up by G05EAF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05EZE. | Class(es): L6b14 | Usage: CALL G05EZF(Z,N,R,NR,IFAIL) | On-line doc: CALL GAMSDOC G05EZF (or @PRT NAG*DOC.G05EZF) | Access: LIB NBS*NAG | See also: G05EAF
- G08AAE** Sign test on two paired samples. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G08AAF. | Class(es): L4b1b | Usage: CALL G08AAE (X, Y, N, IS, N1, P, IFAIL) | On-line doc: CALL GAMSDOC G08AAE (or @PRT NAG*DOC.G08AAE) | Access: LIB NBS*NAG
- G08AAF** Sign test on two paired samples. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G08AAE. | Class(es): L4b1b | Usage: CALL G08AAF (X, Y, N, IS, N1, P, IFAIL) | On-line doc: CALL GAMSDOC G08AAF (or @PRT NAG*DOC.G08AAF) | Access: LIB NBS*NAG
- G08ABE** Wilcoxon matched pairs signed ranks test on two paired samples. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G08ABF. | Class(es): L4b1b | Usage: CALL G08ABE (X, Y, N, W1, W2, W, N1, P, IFAIL) | On-line doc: CALL GAMSDOC G08ABE (or @PRT NAG*DOC.G08ABE) | Access: LIB NBS*NAG
- G08ABF** Wilcoxon matched pairs signed ranks test on two paired samples. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G08ABE. | Class(es): L4b1b | Usage: CALL G08ABF (X, Y, N, W1, W2, W, N1, P, IFAIL) | On-line doc: CALL GAMSDOC G08ABF (or @PRT NAG*DOC.G08ABF) | Access: LIB NBS*NAG
- G08ACE** Median test on two samples of unequal size. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G08ACF. | Class(es): L4b1b | Usage: CALL G08ACE (X, N, N1, W, I1, I2, P, IFAIL) | On-line doc: CALL GAMSDOC G08ACE (or @PRT NAG*DOC.G08ACE) | Access: LIB NBS*NAG
- G08ACF** Median test on two samples of unequal size. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G08ACE. | Class(es): L4b1b | Usage: CALL G08ACF (X, N, N1, W, I1, I2, P, IFAIL) | On-line doc: CALL GAMSDOC G08ACF (or @PRT NAG*DOC.G08ACF) | Access: LIB NBS*NAG
- G08ADE** Mann-Whitney U-test on two samples of unequal size. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G08ADF. | Class(es): L4b1b | Usage: CALL G08ADE (X, N, N1, W, U, P, IFAIL) | On-line doc: CALL GAMSDOC G08ADE (or @PRT NAG*DOC.G08ADE) | Access: LIB NBS*NAG
- G08ADF** Mann-Whitney U-test on two samples of unequal size. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G08ADE. | Class(es): L4b1b | Usage: CALL G08ADF (X, N, N1, W, U, P, IFAIL) | On-line doc: CALL GAMSDOC G08ADF (or @PRT NAG*DOC.G08ADF) | Access: LIB NBS*NAG
- G08AEE** Friedman 2-way analysis of variance on k matched samples. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G08AEF. | Class(es): L7a2a2 | Usage: CALL G08AEE (X, IX, K, N, W1, W2, FR, P, IFAIL) | On-line doc: CALL GAMSDOC G08AEE (or @PRT NAG*DOC.G08AEE) | Access: LIB NBS*NAG
- G08AEF** Friedman 2-way analysis of variance on k matched samples. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G08AEE. | Class(es): L7a2a2 | Usage: CALL G08AEF (X, IX, K, N, W1, W2, FR, P, IFAIL) | On-line doc: CALL GAMSDOC G08AEF (or @PRT NAG*DOC.G08AEF) | Access: LIB NBS*NAG
- G08AFE** Kruskal-Wallis 1-way analysis of variance on k samples of unequal size. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G08AFF. | Class(es): L7a1b | Usage: CALL G08AFE (X, L, LX, K, W, H, P, IFAIL) | On-line doc: CALL GAMSDOC G08AFE (or @PRT NAG*DOC.G08AFE) | Access: LIB NBS*NAG
- G08AFF** Kruskal-Wallis 1-way analysis of variance on k samples of unequal size. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G08AFE. | Class(es): L7a1b | Usage: CALL G08AFF (X, L, LX, K, W, H, P, IFAIL) | On-line doc: CALL GAMSDOC G08AFF (or @PRT NAG*DOC.G08AFF) | Access: LIB NBS*NAG
- G08BAE** Mood's and David's tests on two samples of unequal size. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G08BAF. | Class(es): L4b1b | Usage: CALL G08BAE (X, N, N1, R, ITEST, W, V, PW, PV, IFAIL) | On-line doc: CALL GAMSDOC G08BAE (or @PRT NAG*DOC.G08BAE) | Access: LIB NBS*NAG

- G08BAF** Mood's and David's tests on two samples of unequal size. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G08BAE. | Class(es): L4b1b | Usage: CALL G08BAF (X, N, N1, R, ITEST, W, V, PW, PV, IFAIL) | On-line doc: CALL GAMSDOC G08BAF (or @PRT NAG*DOC.G08BAF) | Access: LIB NBS*NAG
- G08CAE** Kolmogorov-Smirnov one-sample distribution test. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G08CAF. | Class(es): L4a1c | Usage: CALL G08CAE (N, X, NULL, NP, P, NEST, NTYPE, D, PROB, S, IND, IFAIL) | On-line doc: CALL GAMSDOC G08CAE (or @PRT NAG*DOC.G08CAE) | Access: LIB NBS*NAG
- G08CAF** Kolmogorov-Smirnov one-sample distribution test. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G08CAE. | Class(es): L4a1c | Usage: CALL G08CAF (N, X, NULL, NP, P, NEST, NTYPE, D, PROB, S, IND, IFAIL) | On-line doc: CALL GAMSDOC G08CAF (or @PRT NAG*DOC.G08CAF) | Access: LIB NBS*NAG
- G08DAE** Kendall's coefficient of concordance. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G08DAF. | Class(es): L4b1b | Usage: CALL G08DAE (X, IX, K, N, RNK, W, P, IFAIL) | On-line doc: CALL GAMSDOC G08DAE (or @PRT NAG*DOC.G08DAE) | Access: LIB NBS*NAG
- G08DAF** Kendall's coefficient of concordance. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G08DAE. | Class(es): L4b1b | Usage: CALL G08DAF (X, IX, K, N, RNK, W, P, IFAIL) | On-line doc: CALL GAMSDOC G08DAF (or @PRT NAG*DOC.G08DAF) | Access: LIB NBS*NAG
- G13AAE** Carries out non-seasonal and seasonal differencing on a time series. Information which allows the original series to be reconstituted from the differenced series is also produced. This information is required in time series forecasting. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13AAF. | Class(es): L10b | Usage: CALL G13AAE (X, NX, ND, NDS, NS, XD, NXD, IFAIL) | On-line doc: CALL GAMSDOC G13AAE (or @PRT NAG*DOC.G13AAE) | Access: LIB NBS*NAG
- G13AAF** Carries out non-seasonal and seasonal differencing on a time series. Information which allows the original series to be reconstituted from the differenced series is also produced. This information is required in time series forecasting. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13AAE. | Class(es): L10b | Usage: CALL G13AAF (X, NX, ND, NDS, NS, XD, NXD, IFAIL) | On-line doc: CALL GAMSDOC G13AAF (or @PRT NAG*DOC.G13AAF) | Access: LIB NBS*NAG
- G13ABE** Computes the sample autocorrelation function of a time series. It also computes the sample mean, the sample variance and a statistic which may be used to test the hypothesis that the true autocorrelation function is zero. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13ABF. | Class(es): L10c | Usage: CALL G13ABE (X, NX, NK, XM, XV, R, STAT, IFAIL) | On-line doc: CALL GAMSDOC G13ABE (or @PRT NAG*DOC.G13ABE) | Access: LIB NBS*NAG
- G13ABF** Computes the sample autocorrelation function of a time series. It also computes the sample mean, the sample variance and a statistic which may be used to test the hypothesis that the true autocorrelation function is zero. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13ABE. | Class(es): L10c | Usage: CALL G13ABF (X, NX, NK, XM, XV, R, STAT, IFAIL) | On-line doc: CALL GAMSDOC G13ABF (or @PRT NAG*DOC.G13ABF) | Access: LIB NBS*NAG
- G13ACE** Calculates partial autocorrelation coefficients given a set of autocorrelation coefficients. It also calculates the predictor error variance ratios for increasing order of finite lag autoregressive predictor, and the autoregressive parameters associated with the predictor of maximum order. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13ACF. | Class(es): L10c | Usage: CALL G13ACE (R, NK, NL, P, V, AR, NVL, IFAIL) | On-line doc: CALL GAMSDOC G13ACE (or @PRT NAG*DOC.G13ACE) | Access: LIB NBS*NAG
- G13ACF** Calculates partial autocorrelation coefficients given a set of autocorrelation coefficients. It also calculates the predictor error variance ratios for increasing order of finite lag autoregressive predictor, and the autoregressive parameters associated with the predictor of maximum order. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13ACE. | Class(es): L10c | Usage: CALL G13ACF (R, NK, NL, P, V, AR, NVL, IFAIL) | On-line doc: CALL GAMSDOC G13ACF (or @PRT NAG*DOC.G13ACF) | Access: LIB NBS*NAG
- G13ADE** Calculates preliminary estimates of the parameters of an autoregressive moving-average (ARMA) model from an autocorrelation function. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13ADF. | Class(es): L10e1 | Usage: CALL G13ADE (MR, R, NK, XV, NPAR, WA, NWA, PAR, RV, ISF, IFAIL) | On-line doc: CALL GAMSDOC G13ADE (or @PRT NAG*DOC.G13ADE) | Access: LIB NBS*NAG
- G13ADF** Calculates preliminary estimates of the parameters of an autoregressive moving-average (ARMA) model from an autocorrelation function. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13ADE. | Class(es): L10e1 | Usage: CALL G13ADF (MR, R, NK, XV, NPAR, WA, NWA, PAR, RV, ISF, IFAIL) | On-line doc: CALL GAMSDOC G13ADF (or @PRT NAG*DOC.G13ADF) | Access: LIB NBS*NAG
- G13AEE** Iteratively fits seasonal autoregressive-integrated moving-average (ARIMA) model to observed time series using non-linear least squares procedure incorporating backforecasting. Returns parameter estimates, standard errors, residual series, and information for use by G13AGE or G13AHE in forecasting. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13AEF. | Class(es): L10e1 | Usage: CALL G13AEE (MR, PAR, NPAR, C, KFC, X, NX, ICOUNT, EX, EXR, AL, IEX, S, G, IGH, SD, H, IH, ST, IST, NST, PIV, KPIV, NIT, ITC, ZSP, KZSP, ISF, WA, IWA, HC, IFAIL) | On-line doc: CALL GAMSDOC G13AEE (or @PRT NAG*DOC.G13AEE) | Access: LIB NBS*NAG
- G13AEF** Iteratively fits seasonal autoregressive-integrated moving-average (ARIMA) model to observed time series using non-linear least squares procedure incorporating backforecasting. Returns parameter estimates, standard errors, residual series, and information for use by G13AGF or G13AHF in forecasting. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is

- G13AEE.** | Class(es): L10e1 | Usage: CALL G13AEF (MR, PAR, NPAR, C, KFC, X, NX, ICOUNT, EX, EXR, AL, IEX, S, G, IGH, SD, H, IH, ST, IST, NST, PIV, KPIV, NIT, ITC, ZSP, KZSP, ISF, WA, IWA, HC, IFAIL) | On-line doc: CALL GAMSDOC G13AEF (or @PRT NAG*DOC.G13AEF) | Access: LIB NBS*NAG
- G13AFE** Easy-to-use version of G13AEE. Iteratively fits seasonal ARIMA model to observed series using non-linear least squares procedure incorporating backforecasting. Returns parameter estimates, standard errors, residual series, and information for use by G13AGE or G13AHE in forecasting. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13AFF. | Class(es): L10e1 | Usage: CALL G13AFE (MR, PAR, NPAR, C, KFC, X, NX, S, NDF, SD, NPPC, CM, ICM, ST, NST, KPIV, NIT, ITC, ISF, RES, IRES, NRES, IFAIL) | On-line doc: CALL GAMSDOC G13AFE (or @PRT NAG*DOC.G13AFE) | Access: LIB NBS*NAG
- G13AFF** Easy-to-use version of G13AEF. Iteratively fits seasonal ARIMA model to observed series using non-linear least squares procedure incorporating backforecasting. Returns parameter estimates, standard errors, residual series, and information for use by G13AGF or G13AHF in forecasting. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13AFE. | Class(es): L10e1 | Usage: CALL G13AFF (MR, PAR, NPAR, C, KFC, X, NX, S, NDF, SD, NPPC, CM, ICM, ST, NST, KPIV, NIT, ITC, ISF, RES, IRES, NRES, IFAIL) | On-line doc: CALL GAMSDOC G13AFF (or @PRT NAG*DOC.G13AFF) | Access: LIB NBS*NAG
- G13AGE** Accepts new observation of fully specified (by G13AEE or G13AFE) time series and updates "state set" (from G13AEE or G13AFE) information for use in further forecasting. Returns residuals corresponding to the new observations, which may be used in checking that new observations conform to previously fitted model. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13AGF. | Class(es): L10e2 | Usage: CALL G13AGE (ST, NST, MR, PAR, NPAR, C, ANX, NUV, ANEXR, WA, NWA, IFAIL) | On-line doc: CALL GAMSDOC G13AGE (or @PRT NAG*DOC.G13AGE) | Access: LIB NBS*NAG | See also: G13AEE G13AFE
- G13AGF** Accepts new observation of fully specified (by G13AEF or G13AFF) time series and updates "state set" (from G13AEF or G13AFF) information for use in further forecasting. Returns residuals corresponding to the new observations, which may be used in checking that new observations conform to previously fitted model. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13AGE. | Class(es): L10e2 | Usage: CALL G13AGF (ST, NST, MR, PAR, NPAR, C, ANX, NUV, ANEXR, WA, NWA, IFAIL) | On-line doc: CALL GAMSDOC G13AGF (or @PRT NAG*DOC.G13AGF) | Access: LIB NBS*NAG | See also: G13AEF G13AFF
- G13AHE** Produces forecasts of a time series, given model already fitted (by G13AEE or G13AFE). Original observations are not required, since the subroutine uses state set produced originally by G13AEE or G13AFE or updated by G13AGE. Standard errors of the forecasts are also provided. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13AHF. | Class(es): L10e2 | Usage: CALL G13AHE (ST, NST, MR, PAR, NPAR, C, RMS, NFV, FVA, FSD, WA, NWA, IFAIL) | On-line doc: CALL GAMSDOC G13AHE (or @PRT NAG*DOC.G13AHE) | Access: LIB NBS*NAG | See also: G13AEE G13AFE
- G13AHF** Produces forecasts of a time series, given model already fitted (by G13AEF or G13AFF). Original observations are not required, since the subroutine uses state set produced originally by G13AEF or G13AFF or updated by G13AGF. Standard errors of the forecasts are also provided. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13AHE. | Class(es): L10e2 | Usage: CALL G13AHF (ST, NST, MR, PAR, NPAR, C, RMS, NFV, FVA, FSD, WA, NWA, IFAIL) | On-line doc: CALL GAMSDOC G13AHF (or @PRT NAG*DOC.G13AHF) | Access: LIB NBS*NAG | See also: G13AEF G13AFF
- G13AJE** Applies a fully specified seasonal ARIMA model to an observed time series, generates the state set for forecasting and (optionally) derives a specified number of forecasts together with their standard deviations. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13AJF. | Class(es): L10e2 | Usage: CALL G13AJE(MR,PAR,NPAR,C,KFC,X,NX,RMS,ST,IST,NST,NFV,FVA,FSD, IFV,ISF,W, IW,IFAIL) | On-line doc: CALL GAMSDOC G13AJE (or @PRT NAG*DOC.G13AJE) | Access: LIB NBS*NAG
- G13AJF** Applies a fully specified seasonal ARIMA model to an observed time series, generates the state set for forecasting and (optionally) derives a specified number of forecasts together with their standard deviations. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13AJE. | Class(es): L10e2 | Usage: CALL G13AJF(MR,PAR,NPAR,C,KFC,X,NX,RMS,ST,IST,NST,NFV,FVA,FSD, IFV,ISF,W, IW,IFAIL) | On-line doc: CALL GAMSDOC G13AJF (or @PRT NAG*DOC.G13AJF) | Access: LIB NBS*NAG
- G13BAE** Filters a time series by an ARIMA model. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13BAF. | Class(es): L10b | Usage: CALL G13BAE(Y,NY,MR,NMR,PAR,NPAR,CY,WA,NWA,B,NB,IFAIL) | On-line doc: CALL GAMSDOC G13BAE (or @PRT NAG*DOC.G13BAE) | Access: LIB NBS*NAG
- G13BAF** Filters a time series by an ARIMA model. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13BAE. | Class(es): L10b | Usage: CALL G13BAF(Y,NY,MR,NMR,PAR,NPAR,CY,WA,NWA,B,NB,IFAIL) | On-line doc: CALL GAMSDOC G13BAF (or @PRT NAG*DOC.G13BAF) | Access: LIB NBS*NAG
- G13BCE** Calculates cross correlations between two time series. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13BCF. | Class(es): L10g1 | Usage: CALL G13BCE(X,Y,NXY,NL,S,RO,R,STAT,IFAIL) | On-line doc: CALL GAMSDOC G13BCE (or @PRT NAG*DOC.G13BCE) | Access: LIB NBS*NAG
- G13BCF** Calculates cross correlations between two time series. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13BCE. | Class(es): L10g1 | Usage: CALL G13BCF(X,Y,NXY,NL,S,RO,R,STAT,IFAIL) | On-line doc: CALL GAMSDOC G13BCF (or @PRT NAG*DOC.G13BCF) | Access: LIB NBS*NAG
- G13CAE** Calculates the smoothed sample spectrum of a univariate time series using one of four windows - rectangular, Bartlett, Tukey, or Parzen window. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13CAF. | Class(es):

- L10f | Usage: CALL G13CAE(NX,MTX,PX,IW,MW,IC,NC,C,KC,L,LG,NXG,XG,NG,STATS, IFAIL) | On-line doc: CALL GAMSDOC G13CAE (or @PRT NAG*DOC.G13CAE) | Access: LIB NBS*NAG
- G13CAF** Calculates the smoothed sample spectrum of a univariate time series using one of four windows -rectangular, Bartlett, Tukey, or Parzen window. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13CAE. | Class(es): L10f | Usage: CALL G13CAF(NX,MTX,PX,IW,MW,IC,NC,C,KC,L,LG,NXG,XG,NG,STATS, IFAIL) | On-line doc: CALL GAMSDOC G13CAF (or @PRT NAG*DOC.G13CAF) | Access: LIB NBS*NAG
- G13CBE** Calculates the smoothed sample spectrum of a univariate time series using spectral smoothing by the trapezium frequency (Danieli) window. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13CBF. | Class(es): L10f | Usage: CALL G13CBE(NX,MTX,PX,MW,PW,L,KC,LG,XG,NG,STATS,IFAIL) | On-line doc: CALL GAMSDOC G13CBE (or @PRT NAG*DOC.G13CBE) | Access: LIB NBS*NAG
- G13CBF** Calculates the smoothed sample spectrum of a univariate time series using spectral smoothing by the trapezium frequency (Danieli) window. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13CBE. | Class(es): L10f | Usage: CALL G13CBF(NX,MTX,PX,MW,PW,L,KC,LG,XG,NG,STATS,IFAIL) | On-line doc: CALL GAMSDOC G13CBF (or @PRT NAG*DOC.G13CBF) | Access: LIB NBS*NAG
- G13CCE** Calculates the smoothed sample cross spectrum of a bivariate time series using one of four lag windows - rectangular, Bartlett, Tukey, or Parzen window. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13CCF. | Class(es): L10f | Usage: CALL G13CCE(NXY,MTXY,PXY,IW,MW,IS,IC,NC,CXY,CYX,KC,L,NXYG,XG, YG,NG,IFAIL) | On-line doc: CALL GAMSDOC G13CCE (or @PRT NAG*DOC.G13CCE) | Access: LIB NBS*NAG
- G13CCF** Calculates the smoothed sample cross spectrum of a bivariate time series using one of four lag windows - rectangular, Bartlett, Tukey, or Parzen window. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13CCE. | Class(es): L10f | Usage: CALL G13CCF(NXY,MTXY,PXY,IW,MW,IS,IC,NC,CXY,CYX,KC,L,NXYG,XG, YG,NG,IFAIL) | On-line doc: CALL GAMSDOC G13CCF (or @PRT NAG*DOC.G13CCF) | Access: LIB NBS*NAG
- G13CDE** Calculates the smoothed sample cross spectrum of a bivariate time series using spectral smoothing by the trapezium frequency (Danieli) window. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13CDF. | Class(es): L10f | Usage: CALL G13CDE(NXY,MTXY,PXY,MW,IS,PW,L,KC,XG,NG,IFAIL) | On-line doc: CALL GAMSDOC G13CDE (or @PRT NAG*DOC.G13CDE) | Access: LIB NBS*NAG
- G13CDF** Calculates the smoothed sample cross spectrum of a bivariate time series using spectral smoothing by the trapezium frequency (Danieli) window. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13CDE. | Class(es): L10f | Usage: CALL G13CDF(NXY,MTXY,PXY,MW,IS,PW,L,KC,XG,NG,IFAIL) | On-line doc: CALL GAMSDOC G13CDF (or @PRT NAG*DOC.G13CDF) | Access: LIB NBS*NAG
- G13CEE** For a bivariate time series, calculates the cross amplitude spectrum and squared coherency, together with lower and upper bounds from the univariate and bivariate (cross) spectra. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13CEF. | Class(es): L10f | Usage: CALL G13CEE(XG,YG,XYRG,XYIG,NG,STATS,CA,CALW,CAUP,T,SC,SCLW, SCUP,IFAIL) | On-line doc: CALL GAMSDOC G13CEE (or @PRT NAG*DOC.G13CEE) | Access: LIB NBS*NAG
- G13CEF** For a bivariate time series, calculates the cross amplitude spectrum and squared coherency, together with lower and upper bounds from the univariate and bivariate (cross) spectra. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13CEE. | Class(es): L10f | Usage: CALL G13CEF(XG,YG,XYRG,XYIG,NG,STATS,CA,CALW,CAUP,T,SC,SCLW, SCUP,IFAIL) | On-line doc: CALL GAMSDOC G13CEF (or @PRT NAG*DOC.G13CEF) | Access: LIB NBS*NAG
- G13CFE** For a bivariate time series, calculates the gain and phase together with lower and upper bounds from the univariate and bivariate spectra. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13CFF. | Class(es): L10f | Usage: CALL G13CFE(XG,YG,XYRG,XYIG,NG,STATS,GN,GNLW,GNUP,PH,PHLW,PHUP, IFAIL) | On-line doc: CALL GAMSDOC G13CFE (or @PRT NAG*DOC.G13CFE) | Access: LIB NBS*NAG
- G13CFF** For a bivariate time series, calculates the gain and phase together with lower and upper bounds from the univariate and bivariate spectra. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13CFE. | Class(es): L10f | Usage: CALL G13CFF(XG,YG,XYRG,XYIG,NG,STATS,GN,GNLW,GNUP,PH,PHLW,PHUP, IFAIL) | On-line doc: CALL GAMSDOC G13CFF (or @PRT NAG*DOC.G13CFF) | Access: LIB NBS*NAG
- G13CGE** For a bivariate time series, calculates the noise spectrum together with multiplying factors for the bounds and the impulse response function and its standard error, from the univariate and bivariate spectra. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13CGF. | Class(es): L10f | Usage: CALL G13CGE(XG,YG,XYRG,XYIG,NG,STATS,L,N,ER,ERLW,ERUP,RF,RFSE, IFAIL) | On-line doc: CALL GAMSDOC G13CGE (or @PRT NAG*DOC.G13CGE) | Access: LIB NBS*NAG
- G13CGF** For a bivariate time series, calculates the noise spectrum together with multiplying factors for the bounds and the impulse response function and its standard error, from the univariate and bivariate spectra. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13CGE. | Class(es): L10f | Usage: CALL G13CGF(XG,YG,XYRG,XYIG,NG,STATS,L,N,ER,ERLW,ERUP,RF,RFSE, IFAIL) | On-line doc: CALL GAMSDOC G13CGF (or @PRT NAG*DOC.G13CGF) | Access: LIB NBS*NAG
- GAMCDF** Computes the cumulative distribution function value for the gamma distribution with tail length parameter = GAMMA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1g | Usage: CALL GAMCDF(X,GAMMA,CDF) | On-line doc:

CALL GAMSDOC GAMCDF (or @PRT DATAPAC*DOC.GAMCDF) | Access: LIB NBS*DATAPAC

GAMI Incomplete gamma function, = the integral from 0 to x of $(t^{a-1} * e^{-t})dt$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DGAMI. | Class(es): C7e | Usage: Y = GAMI (A,X) | On-line doc: CALL GAMSDOC GAMI (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

GAMIC Complementary incomplete gamma, = the integral from 0 to infinity of $(t^{a-1} * e^{-t})dt$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DGAMIC. | Class(es): C7e | Usage: Y = GAMIC (A,X) | On-line doc: CALL GAMSDOC GAMIC (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

GAMIT Tricomi's incomplete gamma, = $x^{a-1} * \text{incomplete gamma}(a, x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DGAMIT. | Class(es): C7e | Usage: Y = GAMIT (A,X) | On-line doc: CALL GAMSDOC GAMIT (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

GAMLN Computes natural log of Gamma function for non-negative argument. | Portable single precision Fortran subprogram in AMOSLIB sublibrary of CMLIB library. | Class(es): C7a | Usage: Y=GAMLN(X) | On-line doc: CALL GAMSDOC GAMLN (or @PRT CMLIB*DOC.GAMLN/AMOSLIB) | Tests: CMLIB*TEST-SOURCE.GAMLN/AMOSLIB | Access: LIB NBS*CMLIB

GAMMA Gamma function. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DGAMMA. | Class(es): C7a | Usage: Y = GAMMA (X) | On-line doc: CALL GAMSDOC GAMMA (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB | See also: GAMLIM

GAMMA Evaluate the gamma function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): C7a | Usage: X = GAMMA (Y) | On-line doc: CALL GAMSDOC GAMMA (or @PRT IMSL*DOC.GAMMA) | Access: LIB NBS*IMSL

GAMPLT Generates a gamma probability plot with tail length parameter = GAMMA, mean = GAMMA, and standard deviation = $\sqrt{\text{GAMMA}}$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c4g | Usage: CALL GAMPLT(X,N,GAMMA) | On-line doc: CALL GAMSDOC GAMPLT (or @PRT DATAPAC*DOC.GAMPLT) | Access: LIB NBS*DATAPAC

GAMPPF Computes the percent point function value for the gamma distribution with mean = GAMMA and standard deviation = $\sqrt{\text{GAMMA}}$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2g | Usage: CALL GAMPPF(P,GAMMA,PPF) | On-line doc: CALL GAMSDOC GAMPPF (or @PRT DATAPAC*DOC.GAMPPF) | Access: LIB NBS*DATAPAC

GAMR Reciprocal gamma function, = $1 / \text{gamma}(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DGAMR. | Class(es): C7a | Usage: Y = GAMR (X) | On-line doc: CALL GAMSDOC GAMR (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

GAMRAN Generates a random sample of size N from the gamma distribution with tail length parameter = GAMMA, mean = GAMMA and standard deviation = $\sqrt{\text{GAMMA}}$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a7 | Usage: CALL GAMRAN(N,GAMMA,ISTART,X) | On-line doc: CALL GAMSDOC GAMRAN (or @PRT DATAPAC*DOC.GAMRAN) | Access: LIB NBS*DATAPAC

GAUSQ Finds the abscissae and weights for Gauss quadrature on the interval (a,b) for a general weight function with known moments. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DGAUSQ. | Class(es): H2c | Usage: CALL GAUSQ (N,A,B,C,NU,X,W) | On-line doc: CALL GAMSDOC GAUSQ (or @PRT PORT*DOC.GAUSQ) | Access: LIB NBS*PORT

GENBUN Solves certain block tridiagonal systems of lin. equations arising from Hemholtz or Poisson equation in two Cartesian coordinates. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): I2b4b | Usage: CALL GENBUN(NPEROD,N,MPEROD,M,A,B,C,IDIMY,Y,IERROR,W) | On-line doc: CALL GAMSDOC GENBUN (or @PRT CMLIB*DOC.GENBUN/FSHPK) | Tests: CMLIB*TEST-SOURCE.GENBUN/FSHPK | Access: LIB NBS*CMLIB

GEOCDF Computes the geometric cumulative distribution function value at the value X with parameter = P. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1g | Usage: CALL GEOCDF(X,P,CDF) | On-line doc: CALL GAMSDOC GEOCDF (or @PRT DATAPAC*DOC.GEOCDF) | Access: LIB NBS*DATAPAC

GEOPLT Generates a geometric probability plot with parameter P. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c4g | Usage: CALL GEOPLT(X,N,P) | On-line doc: CALL GAMSDOC GEOPLT (or @PRT DATAPAC*DOC.GEOPLT) | Access: LIB NBS*DATAPAC

GEOPPF Computes the percent point function value for the geometric distribution with parameter PPAR. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2g | Usage: CALL GEOPPF(P,PPAR,PPF) | On-line doc: CALL GAMSDOC GEOPPF (or @PRT DATAPAC*DOC.GEOPPF) | Access: LIB NBS*DATAPAC

GEORAN Generates a random sample of size N from the geometric distribution with parameter P. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a7 | Usage: CALL GEORAN(N,P,ISTART,X) | On-line doc: CALL GAMSDOC GEORAN (or @PRT DATAPAC*DOC.GEORAN) | Access: LIB NBS*DATAPAC

GFIT Chi-squared goodness of fit test. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1c | Usage: CALL GFIT (CDF,K,OBS,N,CELLS,COMP,CS,IDF,Q,IER) | On-line doc: CALL GAMSDOC GFIT (or @PRT IMSL*DOC.GFIT) | Access: LIB NBS*IMSL

GGAMR One parameter gamma random deviate generator, and usable as basis for 2 parameter gamma, exponential, chi-squared, chi, beta, t

- and F deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a7 | Usage: CALL GGAMR (DSEED,A,NR,WK,R) | On-line doc: CALL GAMSDOC GGAMR (or @PRT IMSL*DOC.GGAMR) | Access: LIB NBS*IMSL
- GGBN** Binomial random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a2 | Usage: CALL GGBN (DSEED,NR,NIND,P,IR) | On-line doc: CALL GAMSDOC GGBN (or @PRT IMSL*DOC.GGBN) | Access: LIB NBS*IMSL
- GGBNR** Negative binomial random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a14 | Usage: CALL GGBNR (DSEED,K,P,NR,WK,IR) | On-line doc: CALL GAMSDOC GGBNR (or @PRT IMSL*DOC.GGBNR) | Access: LIB NBS*IMSL
- GGBTR** Beta random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a2 | Usage: CALL GGBTR (DSEED,P,Q,NR,R) | On-line doc: CALL GAMSDOC GGBTR (or @PRT IMSL*DOC.GGBTR) | Access: LIB NBS*IMSL
- GGCAY** Cauchy random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a3 | Usage: CALL GGCAY (DSEED,NR,WK,R) | On-line doc: CALL GAMSDOC GGCAY (or @PRT IMSL*DOC.GGCAY) | Access: LIB NBS*IMSL
- GGCHS** Chi-squared random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a3 | Usage: CALL GGCHS (DSEED,N,R,CHI2) | On-line doc: CALL GAMSDOC GGCHS (or @PRT IMSL*DOC.GGCHS) | Access: LIB NBS*IMSL
- GGCOR** Generate a random orthogonal matrix and a random correlation matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6b3 L6b15 | Usage: CALL GGCOR (DSEED, N, E, A, IA, COR, IWK, WK, IER) | On-line doc: CALL GAMSDOC GGCOR (or @PRT IMSL*DOC.GGCOR) | Access: LIB NBS*IMSL
- GGDA** General discrete distribution random deviate generator using alias method. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a7 | Usage: CALL GGDA (DSEED,NR,NDMP,P,IA,WK,IR) | On-line doc: CALL GAMSDOC GGDA (or @PRT IMSL*DOC.GGDA) | Access: LIB NBS*IMSL
- GGDT** General discrete distribution random deviate generator using table lookup method. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a7 | Usage: CALL GGDT (DSEED,NR,L1,IOPT,PF,CP,DEL,LCP,L2,IR,IER) | On-line doc: CALL GAMSDOC GGDT (or @PRT IMSL*DOC.GGDT) | Access: LIB NBS*IMSL
- GGEOT** Geometric random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a7 | Usage: CALL GGEOT (DSEED,NR,P,WK,IR) | On-line doc: CALL GAMSDOC GGEOT (or @PRT IMSL*DOC.GGEOT) | Access: LIB NBS*IMSL
- GGEXN** Exponential random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a5 | Usage: CALL GGEXN (DSEED,XM,NR,R) | On-line doc: CALL GAMSDOC GGEXN (or @PRT IMSL*DOC.GGEXN) | Access: LIB NBS*IMSL
- GGEXT** Random deviate generator for a mixture of two exponentials. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a5 | Usage: CALL GGEXT (DSEED,P,XM1,XM3,NR,R,IER) | On-line doc: CALL GAMSDOC GGEXT (or @PRT IMSL*DOC.GGEXT) | Access: LIB NBS*IMSL
- GGHPR** Hypergeometric random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a8 | Usage: CALL GGHPR (DSEED,N,L,M,NR,WK,IR) | On-line doc: CALL GAMSDOC GGHPR (or @PRT IMSL*DOC.GGHPR) | Access: LIB NBS*IMSL
- GGMTN** Multinomial random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6b13 | Usage: CALL GGMTN (DSEED,NR,NIND,K,P,IR,IR) | On-line doc: CALL GAMSDOC GGMTN (or @PRT IMSL*DOC.GGMTN) | Access: LIB NBS*IMSL
- GGNLG** Log-normal random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a12 | Usage: CALL GGNLG (DSEED,NR,XM,S,R) | On-line doc: CALL GAMSDOC GGNLG (or @PRT IMSL*DOC.GGNLG) | Access: LIB NBS*IMSL
- GGNML** Normal or Gaussian random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a14 | Usage: CALL GGNML (DSEED,NR,R) | On-line doc: CALL GAMSDOC GGNML (or @PRT IMSL*DOC.GGNML) | Access: LIB NBS*IMSL
- GGNO** Generate set of order statistics from normal distribution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a15 L6a14 | Usage: CALL GGNO (DSEED,IFIRST,ILAST,N,R,IER) | On-line doc: CALL GAMSDOC GGNO (or @PRT IMSL*DOC.GGNO) | Access: LIB NBS*IMSL
- GGNPM** Normal random deviate generator via the polar method. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a14 | Usage: CALL GGNPM (DSEED,NR,R) | On-line doc: CALL GAMSDOC GGNPM (or @PRT IMSL*DOC.GGNPM) | Access: LIB NBS*IMSL
- GGNPP** Nonhomogeneous Poisson process generator with rate function lambda(t) - fixed interval, fixed number, or one at a time. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a16 | Usage: CALL GGNPP (DSEED,TL,TU,NUB,FUNLAM,RLAMAX,RLAMIN,IOPT,N,R,IER) | On-line doc: CALL GAMSDOC GGNPP (or @PRT IMSL*DOC.GGNPP) | Access: LIB NBS*IMSL
- GGNQF** Normal random deviate generator. Function form of GGNML. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a14 | Usage: X = GGNQF (DSEED) | On-line doc: CALL GAMSDOC GGNQF (or @PRT IMSL*DOC.GGNQF) | Access: LIB NBS*IMSL

- GGNSM** Multivariate normal random deviate generator with given covariance matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6b14 | Usage: CALL GGNSM (DSEED,NR,K,SIGMA,IR,RVEC,WKVEC,IER) | On-line doc: CALL GAMSDOC GGNSM (or @PRT IMSL*DOC.GGNSM) | Access: LIB NBS*IMSL
- GGPER** Generate a random permutation of the integers 1 to k. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a16 | Usage: CALL GGPER (DSEED,K,IPER) | On-line doc: CALL GAMSDOC GGPER (or @PRT IMSL*DOC.GGPER) | Access: LIB NBS*IMSL
- GGPON** Poisson random deviate generator where the Poisson parameter changes frequently. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a16 | Usage: CALL GGPON (RLAM,DSEED,NR,IR,IER) | On-line doc: CALL GAMSDOC GGPON (or @PRT IMSL*DOC.GGPON) | Access: LIB NBS*IMSL
- GGPOS** Poisson random deviate generator where the Poisson parameter does not change often. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a16 | Usage: CALL GGPOS (RLAM,DSEED,NR,IR,IER) | On-line doc: CALL GAMSDOC GGPOS (or @PRT IMSL*DOC.GGPOS) | Access: LIB NBS*IMSL
- GGSPH** Generation of uniform random deviates from the surface of the unit sphere in 3 or 4 space. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6b21 | Usage: CALL GGSPH (DSEED,NR,IOPT,IZ,Z,IER) | On-line doc: CALL GAMSDOC GGSPH (or @PRT IMSL*DOC.GGSPH) | Access: LIB NBS*IMSL
- GGSRS** Generate a simple random sample from a finite population. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a19 | Usage: CALL GGSRS (DSEED,IOPT,NPOP,IP,MPOP,POP,NSAMP,MSAMP,SAMP,IX, IER) | On-line doc: CALL GAMSDOC GGSRS (or @PRT IMSL*DOC.GGSRS) | Access: LIB NBS*IMSL
- GGSTA** Stable distribution random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a19 | Usage: CALL GGSTA (DSEED,ALPHA,BPRIM,NR,R) | On-line doc: CALL GAMSDOC GGSTA (or @PRT IMSL*DOC.GGSTA) | Access: LIB NBS*IMSL
- GGTAB** Generate a random contingency table with given row and column totals. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6b3 | Usage: CALL GGTAB (DSEED, NROW, NCOL, NRTOT, NCTOT, IND, IIT, ITAB, IWK, IER) | On-line doc: CALL GAMSDOC GGTAB (or @PRT IMSL*DOC.GGTAB) | Access: LIB NBS*IMSL
- GGTRA** Triangular distribution random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a20 | Usage: CALL GGTRA (DSEED,NR,R) | On-line doc: CALL GAMSDOC GGTRA (or @PRT IMSL*DOC.GGTRA) | Access: LIB NBS*IMSL
- GGUBFS** Basic uniform (0,1) random number generator. Function form of GGUBS. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a21 | Usage: X = GGUBFS (DSEED) | On-line doc: CALL GAMSDOC GGUBFS (or @PRT IMSL*DOC.GGUBFS) | Access: LIB NBS*IMSL
- GGUBS** Basic uniform (0,1) pseudo-random number generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a21 | Usage: CALL GGUBS (DSEED,NR,R) | On-line doc: CALL GAMSDOC GGUBS (or @PRT IMSL*DOC.GGUBS) | Access: LIB NBS*IMSL
- GGUBT** Uniform (0,1) pseudo-random number generator using alternate multiplier. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a21 | Usage: CALL GGUBT (DSEED,NR,R) | On-line doc: CALL GAMSDOC GGUBT (or @PRT IMSL*DOC.GGUBT) | Access: LIB NBS*IMSL
- GGUD** Discrete uniform random number generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a21 | Usage: CALL GGUD (DSEED,KN,R,IR) | On-line doc: CALL GAMSDOC GGUD (or @PRT IMSL*DOC.GGUD) | Access: LIB NBS*IMSL
- GGUO** Generate set of order statistics from uniform (0,1) distribution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a15 L6a21 | Usage: CALL GGUO (DSEED,IFIRST,ILAST,N,R,IER) | On-line doc: CALL GAMSDOC GGUO (or @PRT IMSL*DOC.GGUO) | Access: LIB NBS*IMSL
- GGUW** Uniform (0,1) random number generator with shuffling. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a21 | Usage: CALL GGUW (DSEED,NR,IOPT,R) | On-line doc: CALL GAMSDOC GGUW (or @PRT IMSL*DOC.GGUW) | Access: LIB NBS*IMSL
- GGVCR** General continuous distribution random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a7 | Usage: CALL GGVCR (SUBRF,TBL,P1,P2,I3,I4,I5,IT,IOP,DSEED,R,IER) | On-line doc: CALL GAMSDOC GGVCR (or @PRT IMSL*DOC.GGVCR) | Access: LIB NBS*IMSL
- GGVMS** Von Mises random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a22 | Usage: CALL GGVMS (DSEED, C, NR, R) | On-line doc: CALL GAMSDOC GGVMS (or @PRT IMSL*DOC.GGVMS) | Access: LIB NBS*IMSL
- GGWIB** Weibull random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a23 | Usage: CALL GGWIB (DSEED,A,NR,R) | On-line doc: CALL GAMSDOC GGWIB (or @PRT IMSL*DOC.GGWIB) | Access: LIB NBS*IMSL
- GQOIN** Finds the abscissae and weights for Gauss Laguerre quadrature on the interval (0 +infinity). | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DGQOIN. | Class(es): H2c | Usage: CALL GQOIN (N,X,W) | On-line doc: CALL GAMSDOC GQOIN (or @PRT PORT*DOC.GQOIN) | Access: LIB NBS*PORT

- GQM11** Finds the abscissae and weights for Gauss Legendre quadrature on the interval (-1,1). | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DGQM11. | Class(es): H2c | Usage: CALL GQM11 (N,X,W) | On-line doc: CALL GAMSDOC GQM11 (or @PRT PORT*DOC.GQM11) | Access: LIB NBS*PORT
- GTCN** Sample size or number of class intervals determination for chi-squared test applications. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1c | Usage: CALL GTCN (Q,IOPT,B,K,N,IER) | On-line doc: CALL GAMSDOC GTCN (or @PRT IMSL*DOC.GTCN) | Access: LIB NBS*IMSL
- GTD2T** The d-square test. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1c | Usage: CALL GTD2T (COUNT,K,E,CS,STD,Q,IER) | On-line doc: CALL GAMSDOC GTD2T (or @PRT IMSL*DOC.GTD2T) | Access: LIB NBS*IMSL | See also: GTDDU
- GTDDU** D-square tally. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2b | Usage: CALL GTDDU (R,M,COUNT,K,IER) | On-line doc: CALL GAMSDOC GTDDU (or @PRT IMSL*DOC.GTDDU) | Access: LIB NBS*IMSL
- GTMTNT** Moments and standardized moments of uniform random numbers. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1a21 | Usage: CALL GTMTNT (RBAR,R,N,NR,STAT,IER) | On-line doc: CALL GAMSDOC GTMTNT (or @PRT IMSL*DOC.GTMTNT) | Access: LIB NBS*IMSL
- GTNOR** Test for normality of random deviates. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1a14 | Usage: CALL GTNOR (R,N,K,STAT,OBSC,CSOBS,IER) | On-line doc: CALL GAMSDOC GTNOR (or @PRT IMSL*DOC.GTNOR) | Access: LIB NBS*IMSL
- GTPBC** Count of the number of zero bits in a given subset of a real word. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N2 | Usage: CALL GTPBC (I1,I2,R,KZERO) | On-line doc: CALL GAMSDOC GTPBC (or @PRT IMSL*DOC.GTPBC) | Access: LIB NBS*IMSL
- GTPKP** Probability distribution of n elements into two equi-probable states. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1b | Usage: CALL GTPKP (N,P,IER) | On-line doc: CALL GAMSDOC GTPKP (or @PRT IMSL*DOC.GTPKP) | Access: LIB NBS*IMSL
- GTPL** Poker test tally of hand types and statistics. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2b | Usage: CALL GTPL (R,M,I1,I2,IQ,K2,K1,CSOBS,HSAVE,HE,HEH,H,IER) | On-line doc: CALL GAMSDOC GTPL (or @PRT IMSL*DOC.GTPL) | Access: LIB NBS*IMSL
- GTPOK** The poker test. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1c | Usage: CALL GTPOK (HANDS,HE,K1,CSOBS,K2,STAT,IER) | On-line doc: CALL GAMSDOC GTPOK (or @PRT IMSL*DOC.GTPOK) | Access: LIB NBS*IMSL | See also: GTPL
- GTPR** Tally of coordinates of pairs (or lagged pairs) of random numbers. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2b | Usage: CALL GTPR (R,N,K,L,A,IA,IER) | On-line doc: CALL GAMSDOC GTPR (or @PRT IMSL*DOC.GTPR) | Access: LIB NBS*IMSL
- GTPST** Pairs test or Goods serial test. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1d | Usage: CALL GTPST (A,IA,K,CS,Q,STD,IER) | On-line doc: CALL GAMSDOC GTPST (or @PRT IMSL*DOC.GTPST) | Access: LIB NBS*IMSL | See also: GTCN GTPR
- GTRN** Runs test. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1d | Usage: CALL GTRN (RUNS,N,E,CS,Q,STD,IER) | On-line doc: CALL GAMSDOC GTRN (or @PRT IMSL*DOC.GTRN) | Access: LIB NBS*IMSL | See also: GTRTN
- GTRTN** Tally of number of runs up and down. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2b | Usage: CALL GTRTN (R,N,IOPT,WK,RUNS) | On-line doc: CALL GAMSDOC GTRTN (or @PRT IMSL*DOC.GTRTN) | Access: LIB NBS*IMSL
- GTTRT** Tally for triplets test. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2b | Usage: CALL GTTRT (R,N,K,A,IA1,IA2,IER) | On-line doc: CALL GAMSDOC GTTRT (or @PRT IMSL*DOC.GTTRT) | Access: LIB NBS*IMSL
- GTTT** Triplets test. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1d | Usage: CALL GTTT (A,IA1,IA2,K,CS,Q,STD,IER) | On-line doc: CALL GAMSDOC GTTT (or @PRT IMSL*DOC.GTTT) | Access: LIB NBS*IMSL | See also: GTCN GTTRT

H

- H01ABE** Linear programming, simplex algorithm, one iteration. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is H01ABF. | Class(es): G2a1 | Usage: CALL H01ABE (NROW, NCOL, NMVR, NTVR, A, F, IND, IDIM, NOTAB, IRES, IBV, SC, Z) | On-line doc: CALL GAMSDOC H01ABE (or @PRT NAG*DOC.H01ABE) | Access: LIB NBS*NAG
- H01ABF** Linear programming, simplex algorithm, one iteration. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is H01ABE. | Class(es): G2a1 | Usage: CALL H01ABF (NROW, NCOL, NMVR, NTVR, A, F, IND, IDIM, NOTAB, IRES, IBV, SC, Z) | On-line doc: CALL GAMSDOC H01ABF (or @PRT NAG*DOC.H01ABF) | Access: LIB NBS*NAG
- H01ADE** Linear programming, revised simplex method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is H01ADF. | Class(es): G2a1 | Usage: CALL H01ADE (A, MMM, M, N, INEQ, RHS, MIN, MAXIT, MN, MM, M1, M2, CO, IN, II, NBV, D, B, ANS, OPT, NUMIT, IPAR, IFAIL) | On-line doc: CALL GAMSDOC H01ADE (or @PRT NAG*DOC.H01ADE) | Access: LIB NBS*NAG
- H01ADF** Linear programming, revised simplex method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is H01ADE. | Class(es): G2a1 | Usage: CALL H01ADF (A, MMM, M, N, INEQ, RHS, MIN, MAXIT, MN, MM, M1, M2, CO, IN, II, NBV, D, B, ANS, OPT, NUMIT, IPAR, IFAIL) | On-line doc: CALL GAMSDOC H01ADF (or @PRT NAG*DOC.H01ADF) | Access: LIB NBS*NAG
- H01AFE** Find feasible point or vertex which satisfies linear constraints. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is H01AFF. | Class(es): G4d | Usage: CALL H01AFE (N, NGC, AGC, LAGC, IBOUND, BL, BU, LB, ISTATE, LNGC, INITPT, MSGVLV MAXIT, VERTEX,X,NA,R,INDEX,LCON,EL,LEL,QT, LQT, IW,LIW,W,LW,IFAIL) | On-line doc: CALL GAMSDOC H01AFE (or @PRT NAG*DOC.H01AFE) | Access: LIB NBS*NAG
- H01AFF** Find feasible point or vertex which satisfies linear constraints. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is H01AFE. | Class(es): G4d | Usage: CALL H01AFF (N, NGC, AGC, LAGC, IBOUND, BL, BU, LB, ISTATE, LNGC, INITPT, MSGVLV MAXIT, VERTEX,X,NA,R,INDEX,LCON,EL,LEL,QT, LQT, IW,LIW,W,LW,IFAIL) | On-line doc: CALL GAMSDOC H01AFF (or @PRT NAG*DOC.H01AFF) | Access: LIB NBS*NAG
- H01BAE** Linear programming, numerically stable form of simplex method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is H01BAF. | Class(es): G2a1 | Usage: CALL H01BAE (N, NGC, AGC, LAGC, IBOUND, BL, BU, LB, ISTATE, LNGC, INITPT, MSGVLV MAXIT, C,VERTEX,X,F,YMOD,NA,UMIN,R,U,INDEX, LCON, EL,LEL,QT,LQT,IW,LIW,W,L | On-line doc: CALL GAMSDOC H01BAE (or @PRT NAG*DOC.H01BAE) | Access: LIB NBS*NAG
- H01BAF** Linear programming, numerically stable form of simplex method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is H01BAE. | Class(es): G2a1 | Usage: CALL H01BAF (N,NGC,AGC,LAGC,IBOUND,BL, BU,LB,ISTATE,LNGC, INITPT,MSGVLV, MAXIT,C, VERTEX,X,F,YMOD,NA,UMIN,R,U,INDEX,LCON,EL,LEL, QT,LQT,IW,LIW,W,LW,IFAIL) | On-line doc: CALL GAMSDOC H01BAF (or @PRT NAG*DOC.H01BAF) | Access: LIB NBS*NAG
- H02AAE** Quadratic programming, Beale's method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is H02AAF. | Class(es): G2e1 | Usage: CALL H02AAE (MV, N1, MN, ISYM, P, LES1, LES2, MAR, COLK, E) | On-line doc: CALL GAMSDOC H02AAE (or @PRT NAG*DOC.H02AAE) | Access: LIB NBS*NAG
- H02AAF** Quadratic programming, Beale's method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is H02AAE. | Class(es): G2e1 | Usage: CALL H02AAF (MV, N1, MN, ISYM, P, LES1, LES2, MAR, COLK, E) | On-line doc: CALL GAMSDOC H02AAF (or @PRT NAG*DOC.H02AAF) | Access: LIB NBS*NAG
- H02BAE** Integer linear programming, Gomory's method with Wilson's cuts. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is H02BAF. | Class(es): G2c6 | Usage: CALL H02BAE (A, MM, N1, M, N, MAXIT, LL, X, NUMIT, OPT, IFAIL) | On-line doc: CALL GAMSDOC H02BAE (or @PRT NAG*DOC.H02BAE) | Access: LIB NBS*NAG
- H02BAF** Integer linear programming, Gomory's method with Wilson's cuts. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is H02BAE. | Class(es): G2c6 | Usage: CALL H02BAF (A, MM, N1, M, N, MAXIT, LL, X, NUMIT, OPT, IFAIL) | On-line doc: CALL GAMSDOC H02BAF (or @PRT NAG*DOC.H02BAF) | Access: LIB NBS*NAG
- H03ABE** Solves the classical Transportation ("Hitchcock") problem. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is H03ABF. | Class(es): G2b | Usage: CALL H03ABE(KOST,MMM,MA,MB,M,K15,MAXIT,K7,K9,NUMIT,K6,K8,K11, K12, Z,IFAIL) | On-line doc: CALL GAMSDOC H03ABE (or @PRT NAG*DOC.H03ABE) | Access: LIB NBS*NAG
- H03ABF** Solves the classical Transportation ("Hitchcock") problem. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is H03ABE. | Class(es): G2b | Usage: CALL H03ABF (KOST,MMM,MA,MB,M,K15,MAXIT,K7,K9,NUMIT,K6, K8,K11,K12,Z,IFAIL) | On-line doc: CALL GAMSDOC H03ABF (or @PRT NAG*DOC.H03ABF) | Access: LIB NBS*NAG
- HFNCDF** Computes the cumulative distribution function value for the halfnormal distribution with mean = $\sqrt{2/\pi}$ and standard deviation = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1h | Usage: CALL HFNCDF(X,CDF) | On-line doc: CALL GAMSDOC HFNCDF (or @PRT DATAPAC*DOC.HFNCDF) | Access: LIB NBS*DAPAC
- HFNPLT** Generates a halfnormal probability plot with mean = $\sqrt{2/\pi}$ and standard deviation = 1. | Portable single precision Fortran

subprogram in DATAPAC library. | Class(es): L3c4h | Usage: CALL HFNPLT(X,N) | On-line doc: CALL GAMSDOC HFNPLT (or @PRT DATAPAC*DOC.HFNPLT) | Access: LIB NBS*DAPAC

HFNPFF Computes the percent point function value for the halfnormal distribution with mean = $\sqrt{2/\pi}$ and standard deviation = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2h | Usage: CALL HFNPFF(P,PPF) | On-line doc: CALL GAMSDOC HFNPFF (or @PRT DATAPAC*DOC.HFNPFF) | Access: LIB NBS*DAPAC

HFNRAN Generates a random sample of size N from the halfnormal distribution with mean = $\sqrt{2/\pi}$ and standard deviation = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a8 | Usage: CALL HFNRAN(N,ISTART,X) | On-line doc: CALL GAMSDOC HFNRAN (or @PRT DATAPAC*DOC.HFNRAN) | Access: LIB NBS*DAPAC

HFTI Solves linear least squares problem $AX=B$. | Portable single precision Fortran subprogram in FC sublibrary of CMLIB library. | Class(es): D9 | Usage: CALL HFTI(A,MDA,M,N,B, MDB,NB,TAU,KRANK,RNORM,H,G,IP) | On-line doc: CALL GAMSDOC HFTI (or @PRT CMLIB*DOC.HFTI/FC) | Tests: CMLIB*TEST-SOURCE.\$F/FC | Access: LIB NBS*CMLIB

HIST Produces 2 histograms (with differing class widths) of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3a Q1 | Usage: CALL HIST(X,N) | On-line doc: CALL GAMSDOC HIST (or @PRT DATAPAC*DOC.HIST) | Access: LIB NBS*DAPAC

HISTO Produces a histogram and summary statistics. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3a | Usage: CALL HISTO (Y, N, SCRAT, NS) | On-line doc: CALL GAMSDOC HISTO (or @PRT STATLIB*DOC.HISTO) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

HISTOC Produces a histogram and summary statistics, with user control of the number of cells, and of the upper and lower histogram boundaries. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3a | Usage: CALL HISTOC (Y, N, NCELL, YLB, YUB, SCRAT, NS) | On-line doc: CALL GAMSDOC HISTOC (or @PRT STATLIB*DOC.HISTOC) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

HISTOGRAM Prints a histogram of the values in each of one or more vectors, with optional user-specification of the first midpoint and the interval width. | Command in MINITAB Proprietary interactive system. Class(es): L3a Q1 | Usage: HISTogram of C [... and C_j] [first midpoint K, interval width K] | On-line doc: HELP HISTOGRAM (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

HQR Computes eigenvalues of a real upper Hessenberg matrix using the QR method. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c2b | Usage: CALL HQR(NM,N,LOW,IGH,H,WR,WI,IERR) | On-line doc: CALL GAMSDOC HQR (or @PRT CMLIB*DOC.HQR/EISPACK) | Access: LIB NBS*CMLIB

HQR2 Computes eigenvalues and eigenvectors of real upper Hessenberg matrix using QR method. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c2b | Usage: CALL HQR2(NM,N,LOW,IGH,H,WR,WI,Z,IERR) | On-line doc: CALL GAMSDOC HQR2 (or @PRT CMLIB*DOC.HQR2/EISPACK) | Access: LIB NBS*CMLIB

HSTCRT Solves the Helmholtz or Poisson equations in two dimensions in Cartesian coordinates on a staggered grid. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): l2b1a1a | Usage: CALL HSTCRT(A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,ELMBDA,F, IDIMF, PERTRB, IERROR,W) | On-line doc: CALL GAMSDOC HSTCRT (or @PRT CMLIB*DOC.HSTCRT/FSHPK) | Tests: CMLIB*TEST-SOURCE.HSTCRT/FSHPK | Access: LIB NBS*CMLIB

HSTCSP Solves a modified Helmholtz equation in spherical coordinates with axisymmetry using a staggered grid. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): l2b1a1a | Usage: CALL HSTCSP(INTL,A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,ELMBDA,F, IDIMF, PERTRB, IERROR,W) | On-line doc: CALL GAMSDOC HSTCSP (or @PRT CMLIB*DOC.HSTCSP/FSHPK) | Tests: CMLIB*TEST-SOURCE.HSTCSP/FSHPK | Access: LIB NBS*CMLIB

HSTCYL Solves a modified Helmholtz equation in cylindrical coordinates on a staggered grid. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): l2b1a1a | Usage: CALL HSTCYL(A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,ELMBDA,F, IDIMF, PERTRB, IERROR,W) | On-line doc: CALL GAMSDOC HSTCYL (or @PRT CMLIB*DOC.HSTCYL/FSHPK) | Tests: CMLIB*TEST-SOURCE.HSTCYL/FSHPK | Access: LIB NBS*CMLIB

HSTPLR Solves the Helmholtz or Poisson equation in polar coordinates on a staggered grid. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): l2b1a1a | Usage: CALL HSTPLR(A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,ELMBDA,F, IDIMF, PERTRB, IERROR,W) | On-line doc: CALL GAMSDOC HSTPLR (or @PRT CMLIB*DOC.HSTPLR/FSHPK) | Tests: CMLIB*TEST-SOURCE.HSTPLR/FSHPK | Access: LIB NBS*CMLIB

HSTSSP Solves the Helmholtz or Poisson equation in spherical coordinates on the surface of a sphere using a staggered grid. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): l2b1a1a | Usage: CALL HSTSSP(A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,ELMBDA,F, IDIMF, PERTRB, IERROR,W) | On-line doc: CALL GAMSDOC HSTSSP (or @PRT CMLIB*DOC.HSTSSP/FSHPK) | Tests: CMLIB*TEST-SOURCE.HSTSSP/FSHPK | Access: LIB NBS*CMLIB

HTRIB3 Computes eigenvectors of complex Hermitian matrix from eigenvectors of real symmetric tridiagonal matrix output from

HTRID3. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL HTRIB3(NM,N,A,TAU,M,ZR,ZI) | On-line doc: CALL GAMSDOC HTRIB3 (or @PRT CMLIB*DOC.HTRIB3/EISPACK) | Access: LIB NBS*CMLIB | See also: HTRID3

HTRIBK Forms eigenvectors of complex Hermitian matrix from eigenvectors of real symmetric tridiagonal matrix output from HTRIDI. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL HTRIBK(NM,N,AR,AI,TAU,M,ZR,ZI) | On-line doc: CALL GAMSDOC HTRIBK (or @PRT CMLIB*DOC.HTRIBK/EISPACK) | Access: LIB NBS*CMLIB | See also: HTRIDI

HTRID3 Reduces complex Hermitian (packed) matrix to real symmetric tridiagonal matrix by unitary similarity transformations. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1b1 | Usage: CALL HTRID3(NM,N,A,D,E,E2,TAU) | On-line doc: CALL GAMSDOC HTRID3 (or @PRT CMLIB*DOC.HTRID3/EISPACK) | Access: LIB NBS*CMLIB

HTRIDI Reduces complex Hermitian matrix to real symmetric tridiagonal matrix using unitary similarity transformations. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1b1 | Usage: CALL HTRIDI(NM,N,AR,AI,D,E,E2,TAU) | On-line doc: CALL GAMSDOC HTRIDI (or @PRT CMLIB*DOC.HTRIDI/EISPACK) | Access: LIB NBS*CMLIB

HUMSL Minimizes a general unconstrained objective function using (analytic) gradient and Hessian provided by the user. | Portable single precision Fortran subprogram in NL2SN sublibrary of CMLIB library. Double precision version is DHUMSL. | Class(es): G1b1c | Usage: CALL HUMSL(N,D,X,CALCF,CALCGH,IV,LIV,LV,V,UIPARM,URPARM,UFPARM) | On-line doc: CALL GAMSDOC HUMSL (or @PRT CMLIB*DOC.HUMSL/NL2SN) | Tests: CMLIB*TEST-SOURCE.\$F2/NL2SN, CMLIB*TEST-SOURCE.\$Q2/NL2SN | Access: LIB NBS*CMLIB

HW3CRT Solves the Helmholtz or Poisson equation in three dimensions using Cartesian coordinates. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): I2b1a1a | Usage: CALL HW3CRT(XS,XF,L,LBDCND,BDXS,BDXF,YS,YF,M,MBDCND,BDYS,BDYF, ZS,ZF,N,NBDCND,BDZS,BDZF,ELMBDA,LDIMF,MDIMF,F,PERTRB,IERROR,W) | On-line doc: CALL GAMSDOC HW3CRT (or @PRT CMLIB*DOC.HW3CRT/FSHPK) | Tests: CMLIB*TEST-SOURCE.HW3CRT/FSHPK | Access: LIB NBS*CMLIB

HWSCRT Solves the Helmholtz or Poisson equation in two dimensions in Cartesian coordinates. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): I2b1a1a | Usage: CALL HWSCRT(A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,ELMBDA,F, IDIMF, PERTRB, IERROR,W) | On-line doc: CALL GAMSDOC HWSCRT (or @PRT CMLIB*DOC.HWSCRT/FSHPK) | Tests: CMLIB*TEST-SOURCE.HWSCRT/FSHPK | Access: LIB NBS*CMLIB

HWSCSP Solves a modified Helmholtz equation in spherical coordinates with axisymmetry. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): I2b1a1a | Usage: CALL HWSCSP(INTL,TS,TF,M,MBDCND,BDTS,BDTF,RS,RF,N,NBDCND,BDC, BDD,ELMBDA,F,IDIMF, PERTRB,IERROR,W) | On-line doc: CALL GAMSDOC HWSCSP (or @PRT CMLIB*DOC.HWSCSP/FSHPK) | Tests: CMLIB*TEST-SOURCE.HWSCSP/FSHPK | Access: LIB NBS*CMLIB

HWSCYL Solves a modified Helmholtz equation in cylindrical coordinates. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): I2b1a1a | Usage: CALL HWSCYL(A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,ELMBDA,F, IDIMF, PERTRB, IERROR,W) | On-line doc: CALL GAMSDOC HWSCYL (or @PRT CMLIB*DOC.HWSCYL/FSHPK) | Tests: CMLIB*TEST-SOURCE.HWSCYL/FSHPK | Access: LIB NBS*CMLIB

HSWPLR Solves the Helmholtz or Poisson equation in polar coordinates. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): I2b1a1a | Usage: CALL HSWPLR(A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,ELMBDA,F, IDIMF, PERTRB, IERROR,W) | On-line doc: CALL GAMSDOC HSWPLR (or @PRT CMLIB*DOC.HSWPLR/FSHPK) | Tests: CMLIB*TEST-SOURCE.HSWPLR/FSHPK | Access: LIB NBS*CMLIB

HWSSSP Solves the Helmholtz or Poisson equation in spherical coordinates on the surface of a sphere. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): I2b1a1a | Usage: CALL HWSSSP(TS,TF,M,MBDCND,BDTS,BDTF,PS,PF,N,NBDCND,BDPS,BDPF, ELMBDA,F,IDIMF, PERTRB,IERROR,W) | On-line doc: CALL GAMSDOC HWSSSP (or @PRT CMLIB*DOC.HWSSSP/FSHPK) | Tests: CMLIB*TEST-SOURCE.HWSSSP/FSHPK | Access: LIB NBS*CMLIB

I

- IIMACH** Provides integer machine dependent information, e.g. largest integer. | Portable single precision Fortran subprogram in MACHCONST sublibrary of CMLIB library. | Class(es): R1 | Usage: J=IIMACH(I) | On-line doc: CALL GAMSDOC IIMACH (or @PRT CMLIB*DOC.IIMACH/MACHCONST) | Tests: CMLIB*TEST-SOURCE.\$Q/MACHCONST | Access: LIB NBS*CMLIB
- IIMACH** Provides the integer constants required to adapt PORT library programs to individual computers. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): R1 | Usage: I = IIMACH (I) | On-line doc: CALL GAMSDOC IIMACH (or @PRT PORT*DOC.IIMACH) | Access: LIB NBS*PORT
- IASPEC** Computes the integrated sample periodogram of a series (not recommended for long series because of the algorithm used). | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L10f | Usage: CALL IASPEC (Y, N, SCRAT, NS) | On-line doc: CALL GAMSDOC IASPEC (or @PRT STATLIB*DOC.IASPEC) | Tests: STATLIB*TEST.DEMO4 | Access: LIB NBS*STATLIB
- IBCCCU** Bicubic spline two-dimensional coefficient calculator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): E2a | Usage: CALL IBCCCU(F,X,NX,Y,NY,C,IC,WK,IER) | On-line doc: CALL GAMSDOC IBCCCU (or @PRT IMSL*DOC.IBCCCU) | Access: LIB NBS*IMSL | See also: IBCEVL DBCEVL
- IBCEVL** Evaluation of a bicubic spline. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): E3 K6 | Usage: CALL IBCEVL (X,NX,Y,NY,C,IC,XL,YL,FL,IER) | On-line doc: CALL GAMSDOC IBCEVL (or @PRT IMSL*DOC.IBCEVL) | Access: LIB NBS*IMSL
- IBCIEU** Bicubic spline two-dimensional interpolator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): E2a | Usage: CALL IBCIEU (F,IFD,X,NX,Y,NY,XL,NXL,YL,NYL,FL,IFLD,WK,IER) | On-line doc: CALL GAMSDOC IBCIEU (or @PRT IMSL*DOC.IBCIEU) | Access: LIB NBS*IMSL
- ICAMAX** Find smallest index of maximum magnitude component of a complex vector. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. | Class(es): D1a2 | Usage: IMAX = ICAMAX(N,CX,INCX) | On-line doc: CALL GAMSDOC ICAMAX (or @PRT CMLIB*DOC.ICAMAX/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- ICEIL** Finds the smallest integer greater than or equal to x. Input is real, output is integer. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): C1 | Usage: I = ICEIL (X) | On-line doc: CALL GAMSDOC ICEIL (or @PRT PORT*DOC.ICEIL) | Access: LIB NBS*PORT
- ICSCCU** Cubic spline interpolation (easy-to-use version). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): E1a | Usage: CALL ICSCCU (X,Y,NX,C,IC,IER) | On-line doc: CALL GAMSDOC ICSCCU (or @PRT IMSL*DOC.ICSCCU) | Access: LIB NBS*IMSL | See also: ICSEVU DCSEVU
- ICSEVU** Evaluation of a cubic spline. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): E3 K6 | Usage: CALL ICSEVU (X,Y,NX,C,IC,U,S,M,IER) | On-line doc: CALL GAMSDOC ICSEVU (or @PRT IMSL*DOC.ICSEVU) | Access: LIB NBS*IMSL
- ICSFKU** Least squares approximation by cubic splines fixed knots. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): K1a1a1 | Usage: CALL ICSFKU (X,F,NX,MODE,XK,NXK,Y,C,IC,ERROR,WK,IER) | On-line doc: CALL GAMSDOC ICSFKU (or @PRT IMSL*DOC.ICSFKU) | Access: LIB NBS*IMSL | See also: ICSEVU DCSEVU
- ICSICU** Interpolatory approximation by cubic splines with arbitrary second derivative end conditions. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): E1a | Usage: CALL ICSICU (X,Y,NX,BPAR,C,IC,IER) | On-line doc: CALL GAMSDOC ICSICU (or @PRT IMSL*DOC.ICSIKU) | Access: LIB NBS*IMSL | See also: ICSEVU DCSEVU
- ICSMOU** One-dimensional data smoothing by error detection. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): K5 L10b | Usage: CALL ICSMOU (X,Y,NX,DIS,SC,MAXIT,WK,IER) | On-line doc: CALL GAMSDOC ICSMOU (or @PRT IMSL*DOC.ICSMOU) | Access: LIB NBS*IMSL
- ICSPEC** Displays plots of the integrated sample phase and co-spectra for a pair of series (not recommended for long series because of the algorithm used). | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L10g1 | Usage: CALL ICSPEC (Y1, Y2, N, SCRAT, NS) | On-line doc: CALL GAMSDOC ICSPEC (or @PRT STATLIB*DOC.ICSPEC) | Tests: STATLIB*TEST.DEMO4 | Access: LIB NBS*STATLIB
- ICSPLN** Cubic spline interpolation with periodic end conditions. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): E1a | Usage: CALL ICSPLN (X,Y,NX,C,IC,WK,IER) | On-line doc: CALL GAMSDOC ICSPLN (or @PRT IMSL*DOC.ICSPN) | Access: LIB NBS*IMSL | See also: ICSEVU DCSEVU
- ICSSCU** Cubic spline data smoother. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): K5 K1a1a1 | Usage: CALL ICSSCU (X,F,DF,NX,SM,Y,C,IC,WK,IER) | On-line doc: CALL GAMSDOC ICSSCU (or @PRT IMSL*DOC.ICSSCU) | Access: LIB NBS*IMSL | See also: ICSEVU DCSEVU
- ICSSCV** Cubic spline data smoother (easy-to-use version). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): K5 K1a1a1 | Usage: CALL ICSSCV (X,F,NX,Y,C,IC,IJOB,WK,IER) | On-line doc: CALL GAMSDOC ICSSCV (or @PRT IMSL*DOC.ICSSCV) | Access: LIB NBS*IMSL | See also: ICSEVU DCSEVU
- ICSVKU** Least squares approximation by cubic splines - variable knots. | Proprietary single precision Fortran subprogram in IMSL library. |

Class(es): K1a1a1 | Usage: CALL ICSVKU (X,F,NX,XK,NXK,Y,C,IC,EPROR,WK,IER) | On-line doc: CALL GAMSDOC ICSVKU (or @PRT IMSL*DOC.ICSVKU) | Access: LIB NBS*IMSL | See also: ICSEVU DCSEVU

IDAMAX Find smallest index of maximum magnitude component of a double precision vector. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. Single precision version is ISAMAX. | Class(es): D1a2 | Usage: IMAX = IDAMAX(N,DX,INCX) | On-line doc: CALL GAMSDOC IDAMAX (or @PRT CMLIB*DOC.IDAMAX/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

IDCEIL Finds the smallest integer greater than or equal to x. Input is double precision output is integer. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): C1 | Usage: I = IDCEIL (X) | On-line doc: CALL GAMSDOC IDCEIL (or @PRT PORT*DOC.IDCEIL) | Access: LIB NBS*PORT

IDFLR Finds the largest integer less than or equal to x. Input is double precision, output is integer. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is IFLR. | Class(es): C1 | Usage: I = IDFLR (X) | On-line doc: CALL GAMSDOC IDFLR (or @PRT PORT*DOC.IDFLR) | Access: LIB NBS*PORT

IDLUMB Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points for B-spline use. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ILUMB. | Class(es): E3 K6 | Usage: I = IDLUMB (XB,NXB,N,K,NX) | On-line doc: CALL GAMSDOC IDLUMB (or @PRT PORT*DOC.IDLUMB) | Access: LIB NBS*PORT

IDLUMD Given a basic mesh, this subdivides each interval into the same number or uniformly spaced points for B-spline use. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ILUMD. | Class(es): E3 K6 | Usage: I = IDLUMD (XB,NXB,N,NX) | On-line doc: CALL GAMSDOC IDLUMD (or @PRT PORT*DOC.IDLUMD) | Access: LIB NBS*PORT

IDMNPB Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each mesh interval. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): E3 K6 | Usage: I = IDMNPB (X,NX,N,K,NT) | On-line doc: CALL GAMSDOC IDMNPB (or @PRT PORT*DOC.IDMNPB) | Access: LIB NBS*PORT

IDPUMB Given a basic mesh, this subdivides each interval. Number of points per interval can vary, but uniform in each subdivision. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is IPUMB. | Class(es): E3 K6 | Usage: I = IDPUMB (XB,NXB,NA,K,NX) | On-line doc: CALL GAMSDOC IDPUMB (or @PRT PORT*DOC.IDPUMB) | Access: LIB NBS*PORT

IDPUMD Given a basic mesh, this subdivides each interval with a variable number of points. Points are uniform in each interval. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is IPUMD. | Class(es): E3 K6 | Usage: I = IDPUMD (XB,NXB,NA,NX) | On-line doc: CALL GAMSDOC IDPUMD (or @PRT PORT*DOC.IDPUMD) | Access: LIB NBS*PORT

IDUMB Given interval endpoints, this generates a uniform mesh for B-spline use. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is IUMB. | Class(es): E3 K6 | Usage: I = IDUMB (A,B,NAB,K,NX) | On-line doc: CALL GAMSDOC IDUMB (or @PRT PORT*DOC.IDUMB) | Access: LIB NBS*PORT

IDUMD Given interval endpoints, this generates a uniform mesh. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is LUMD. | Class(es): E3 K6 | Usage: I = IDUMD (A,B,NAB) | On-line doc: CALL GAMSDOC IDUMD (or @PRT PORT*DOC.IDUMD) | Access: LIB NBS*PORT

IFLR Finds the largest integer less than or equal to x. Input is real output is integer. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is IDFLR. | Class(es): C1 | Usage: I = IFLR (X) | On-line doc: CALL GAMSDOC IFLR (or @PRT PORT*DOC.IFLR) | Access: LIB NBS*PORT

IFLSQ Least squares approximation with user supplied functions. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): K1a1a3 | Usage: CALL IFLSQ (F,X,Y,M,A,N,WK,IER) | On-line doc: CALL GAMSDOC IFLSQ (or @PRT IMSL*DOC.IFLSQ) | Access: LIB NBS*IMSL

ILUMB Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points for B-spline use. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is IDLUMB. | Class(es): E3 K6 | Usage: I = ILUMB (XB,NXB,N,K,NX) | On-line doc: CALL GAMSDOC ILUMB (or @PRT PORT*DOC.IDLUMB) | Access: LIB NBS*PORT

ILUMD Given a basic mesh, this subdivides each interval into the same number or uniformly spaced points for B-spline use. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is IDLUMD. | Class(es): E3 K6 | Usage: I = ILUMD (XB,NXB,N,NX) | On-line doc: CALL GAMSDOC ILUMD (or @PRT PORT*DOC.IDLUMD) | Access: LIB NBS*PORT

IMNPB Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each mesh interval. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): E3 K6 | Usage: I = IMNPB (X,NX,N,K,NT) | On-line doc: CALL GAMSDOC IMNPB (or @PRT PORT*DOC.IMNPB) | Access: LIB NBS*PORT

IMTQL1 Computes eigenvalues of symmetric tridiagonal matrix using implicit QL method. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 D4c2a | Usage: CALL IMTQL1(N,D,E,IERR) | On-line doc: CALL GAMSDOC IMTQL1 (or @PRT CMLIB*DOC.IMTQL1/EISPACK) | Access: LIB NBS*CMLIB

IMTQL2 Computes eigenvalues and eigenvectors of symmetric tridiagonal matrix using implicit QL method. | Portable single precision Fortran

subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 D4c2a | Usage: CALL IMTQL2(NM,N,D,E,Z,IERR) | On-line doc: CALL GAMSDOC IMTQL2 (or @PRT CMLIB*DOC.IMTQL2/EISPACK) | Access: LIB NBS*CMLIB

IMTQLV Computes eigenvalues of symmetric tridiagonal matrix using implicit QL method. Eigenvectors may be computed later. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 D4c2a | Usage: CALL IMTQLV(N,D,E,E2,W,IND,IERR,RV1) | On-line doc: CALL GAMSDOC IMTQLV (or @PRT CMLIB*DOC.IMTQLV/EISPACK) | Access: LIB NBS*CMLIB

INTRV Computes the index into a knot or breakpoint sequence corresponding to a given point X. | Portable single precision Fortran subprogram in BSPLINE sublibrary of CMLIB library. Double precision version is DINTRV. | Class(es): E3 K6 | Usage: CALL INTRV(KT,LXT,X,ILD,ILEFT,MFLAG) | On-line doc: CALL GAMSDOC INTRV (or @PRT CMLIB*DOC.INTRV/BSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/BSPLINE | Access: LIB NBS*CMLIB

INTRVD Finds the interval in a double precision array to which a double precision element belongs. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is INTRVR. | Class(es): N5b | Usage: 1 = INTRVD (N, D, DD) | On-line doc: CALL GAMSDOC INTRVD (or @PRT PORT*DOC.INTRVD) | Access: LIB NBS*PORT

INTRVI Finds the interval in an integer array to which an integer element belongs. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): N5a | Usage: 1 = INTRVI (N, I, II) | On-line doc: CALL GAMSDOC INTRVI (or @PRT PORT*DOC.INTRVI) | Access: LIB NBS*PORT

INTRVR Finds the interval in a real array to which a real element belongs. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is INTRVD. | Class(es): N5b | Usage: 1 = INTRVR (N, R, RR) | On-line doc: CALL GAMSDOC INTRVR (or @PRT PORT*DOC.INTRVR) | Access: LIB NBS*PORT

INVAR1 Interactive packages for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results. Line printer graphics only. | Portable stand-alone program using INVAR command language. | Class(es): L8g K1b1a1 K1b1a2 | Usage: -0- | On-line doc: CALL GAMSDOC INVAR1 (or @PRT INVAR*DOC.INVAR1) | Access: CALL SCD*CTSLIB. <program name> (in CTS)

INVAR2 Interactive packages for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results and DISSPLA graphics. | Portable stand-alone program using INVAR command language. | Class(es): L8g K1b1a1 K1b1a2 | Usage: -0- | On-line doc: CALL GAMSDOC INVAR2 (or @PRT INVAR*DOC.INVAR2) | Access: CALL SCD*CTSLIB. <program name> (in CTS)

INVIT Computes eigenvectors of upper Hessenberg (real) matrix associated with specified eigenvalues by inverse iteration. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c2b | Usage: CALL INVIT(NM,N,A,WR,W1,SELECT,MM,M,Z,IERR,RM1,RV1,RV2) | On-line doc: CALL GAMSDOC INVIT (or @PRT CMLIB*DOC.INVIT/EISPACK) | Access: LIB NBS*CMLIB

IPUMB Given a basic mesh, this subdivides each interval. Number of points per interval can vary, but uniform in each subdivision. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is IDPUMB. | Class(es): E3 K6 | Usage: 1 = IPUMB (XB,NXB,NA,K,NX) | On-line doc: CALL GAMSDOC IPUMB (or @PRT PORT*DOC.IPUMB) | Access: LIB NBS*PORT

IPUMD Given a basic mesh, this subdivides each interval with a variable number of points. Points are uniform in each interval. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is IDPUMD. | Class(es): E3 K6 | Usage: 1 = IPUMD (XB,NXB,NA,NX) | On-line doc: CALL GAMSDOC IPUMD (or @PRT PORT*DOC.IPUMD) | Access: LIB NBS*PORT

IQHSCU Visually pleasing interpolant of one dimensional data via piecewise cubic Hermite function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): E1a | Usage: CALL IQHSCU (X,Y,NX,C,IC,IER) | On-line doc: CALL GAMSDOC IQHSCU (or @PRT IMSL*DOC.IQHSCU) | Access: LIB NBS*IMSL

IQHSCV Smooth surface fitting with irregularly distributed data points. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): e2b | Usage: CALL IQHSCV (XD,YD,ZD,ND,X1,NX1,Y1,NY1,Z1,IZ1,IWK,WK,IER) | On-line doc: CALL GAMSDOC IQHSCV (or @PRT IMSL*DOC.IQHSCV) | Access: LIB NBS*IMSL

IQHSCV Smooth surface fitting with irregularly distributed data points. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): E2b | Usage: CALL IQHSCV (XD,YD,ZD,ND,X1,NX1,Y1,NY1,Z1,IZ1,IWK,WK,IER) | On-line doc: CALL GAMSDOC IQHSCV (or @PRT IMSL*DOC.IQHSCV) | Access: LIB NBS*IMSL

IRANDOM Generates K pseudo-random integers in a specified interval. | Command in MINITAB Proprietary interactive system. | Class(es): L6a9 | Usage: IRANDOM K random integers between K and K, put into C | On-line doc: HELP IRANDOM (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

IRATCU Rational weighted Chebyshev approximation of a continuous function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): K2 | Usage: CALL IRATCU (F,PHI,G,A,B,L,M,P,Q,WK,IER) | On-line doc: CALL GAMSDOC IRATCU (or @PRT IMSL*DOC.IRATCU) | Access: LIB NBS*IMSL

ISAMAX Find smallest index of maximum magnitude component of a single precision vector. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Double precision version is IDAMAX. | Class(es): D1a2 | Usage: IMAX = ISAMAX(N,SX,INCX) |

On-line doc: CALL GAMSDOC ISAMAX (or @PRT CMLIB*DOC.ISAMAX/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

ISAMIN Find the smallest index of the minimum magnitude component of a real vector. | Portable single precision Fortran subprogram in XBLAS sublibrary of CMLIB library. | Class(es): D1a2 | Usage: I=ISAMIN(N,SX,INCX) | On-line doc: CALL GAMSDOC ISAMIN (or @PRT CMLIB*DOC.ISAMIN/XBLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/XBLAS | Access: LIB NBS*CMLIB

ISMAX Find the smallest index of the maximum component of a real vector. | Portable single precision Fortran subprogram in XBLAS sublibrary of CMLIB library. | Class(es): D1a2 | Usage: I=ISMAX(N,SX,INCX) | On-line doc: CALL GAMSDOC ISMAX (or @PRT CMLIB*DOC.ISMAX/XBLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/XBLAS | Access: LIB NBS*CMLIB

ISMIN Find the smallest index of the minimum component of a real vector. | Portable single precision Fortran subprogram in XBLAS sublibrary of CMLIB library. | Class(es): D1a2 | Usage: I=ISMIN(N,SX,INCX) | On-line doc: CALL GAMSDOC ISMIN (or @PRT CMLIB*DOC.ISMIN/XBLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/XBLAS | Access: LIB NBS*CMLIB

ISORT Sorts an integer array in either increasing or decreasing order. Optionally another integer array can be carried along. | Portable single precision Fortran subprogram in SSORT sublibrary of CMLIB library. | Class(es): N6a2a | Usage: CALL ISORT(IX,IY,N,KFLAG) | On-line doc: CALL GAMSDOC ISORT (or @PRT CMLIB*DOC.ISORT/SSORT) | Tests: CMLIB*TEST-SOURCE.\$Q/SSORT | Access: LIB NBS*CMLIB

ISTKGT Allocates (gets) an array from the storage stack for PORT library programs. | Proprietary mixed precision Fortran subprogram in PORT library. | Class(es): N4 | Usage: I = ISTKGT (NITEMS,ITYPE) | On-line doc: CALL GAMSDOC ISTKGT (or @PRT PORT*DOC.ISTKGT) | Access: LIB NBS*PORT

ISTKIN Initialize the length of the dynamic storage stack for PORT library programs. | Proprietary mixed precision Fortran subprogram in PORT library. | Class(es): N4 | Usage: CALL ISTKIN (NITEMS,ITYPE) | On-line doc: CALL GAMSDOC ISTKIN (or @PRT PORT*DOC.ISTKIN) | Access: LIB NBS*PORT

ISTKMD Changes size of last stack allocation for PORT library programs. | Proprietary mixed precision Fortran subprogram in PORT library. | Class(es): N4 | Usage: I = ISTKMD (NITEMS) | On-line doc: CALL GAMSDOC ISTKMD (or @PRT PORT*DOC.ISTKMD) | Access: LIB NBS*PORT

ISTKQU Returns the number of available items that remain in the stack for PORT library programs. | Proprietary mixed precision Fortran subprogram in PORT library. | Class(es): N4 | Usage: I = ISTKQU (ITYPE) | On-line doc: CALL GAMSDOC ISTKQU (or @PRT PORT*DOC.ISTKQU) | Access: LIB NBS*PORT

ISTKRL Releases the last storage allocations requested for PORT library programs. | Proprietary mixed precision Fortran subprogram in PORT library. | Class(es): N4 | Usage: CALL ISTKRL (NUMBER) | On-line doc: CALL GAMSDOC ISTKRL (or @PRT PORT*DOC.ISTKRL) | Access: LIB NBS*PORT

ISTKST Returns information on the status of the stack for PORT library programs. | Proprietary mixed precision Fortran subprogram in PORT library. | Class(es): N4 | Usage: I = ISTKST (NFACT) | On-line doc: CALL GAMSDOC ISTKST (or @PRT PORT*DOC.ISTKST) | Access: LIB NBS*PORT

IUMB Given interval endpoints, this generates a uniform mesh for B-spline use. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is IDUMB. | Class(es): E3 K6 | Usage: I = IUMB (A,B,NAB,K,NX) | On-line doc: CALL GAMSDOC IUMB (or @PRT PORT*DOC.IUMB) | Access: LIB NBS*PORT

IUMD Given interval endpoints, this generates a uniform mesh. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is IDUMD. | Class(es): E3 K6 | Usage: I = IUMD (A,B,NAB) | On-line doc: CALL GAMSDOC IUMD (or @PRT PORT*DOC.IUMD) | Access: LIB NBS*PORT

J

- JCG** Iterative solution of large sparse systems of linear equations. Jacobi method, conjugate gradient acceleration, adaptive parameter selection. | Portable single precision Fortran subprogram in ITPACK sublibrary of MATHWARE library. | Class(es): D2b4 I2b4b D2a4 | Usage: CALL JCG(N,IA,JA,A,RHS,U,IWKSP,NW,WKSP,IPARM,RPARM,IER) | On-line doc: @PRT,S MATHWARE*ITPACK.DOCUMENT | Access: See individual sublibrary documentation
- JOIN** Merges constants and/or vectors into vectors. | Command in MINITAB Proprietary interactive system. Class(es): L2 | Usage: JOIN E to the bottom of E [to the bottom of E,..., to E] put into C | On-line doc: HELP JOIN (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- JSI** Iterative solution of large sparse systems of linear equations. Jacobi method, Chebyshev acceleration, adaptive parameter selection. | Portable single precision Fortran subprogram in ITPACK sublibrary of MATHWARE library. | Class(es): D2b4 I2b4b D2a4 | Usage: CALL JSI(N,IA,JA,A,RHS,U,IWKSP,NW,WKSP,IPARM,RPARM,IER) | On-line doc: @PRT,S MATHWARE*ITPACK.DOCUMENT | Access: See individual sublibrary documentation

K

KRUSKAL-WALLIS Perform Kruskal-Wallis test, based on ranks, of the null hypothesis that there is no difference among K population locations against the alternative of at least one difference. (This is a K-sample generalization of the Mann-Whitney-Wilcoxon test and is a nonparametric alternative to one-way ANOVA.). | Command in MINITAB Proprietary interactive system. Class(es): L4b1b. | Usage: KRUSkal-Wallis test for data in C, subscripts in C | On-line doc: HELP KRUSKAL-WALLIS (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

L

- L2SFF** Obtains a weighted least square expansion of a known function in terms of B-splines of order K, at given mesh points. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DL2SFF. | Class(es): K1a1a1 | Usage: CALL L2SFF (FW,K,T,NT,A) | On-line doc: CALL GAMSDOC L2SFF (or @PRT PORT*DOC.L2SFF) | Access: LIB NBS*PORT | See also: SPLNE SPLND SPLNI SPLN1 SPLN2 EEBSF EEBSI EESFF EESFI
- L2SFH** Obtains a weighted least square expansion of a known function in and its derivatives in terms of B-splines of order K at given mesh points. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DL2SFH. | Class(es): K1a1a1 | Usage: CALL L2SFH (FW,MD,K,T,NT,A) | On-line doc: CALL GAMSDOC L2SFH (or @PRT PORT*DOC.L2SFH) | Access: LIB NBS*PORT | See also: SPLNE SPLND SPLNI SPLN1 SPLN2
- LAG** Computes lagged observations in a time series. | Command in MINITAB Proprietary interactive system. Class(es): L10a | Usage: LAG [K] data in C, put into C | On-line doc: HELP LAG (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- LAMCDF** Computes the cumulative distribution function value for the (Tukey) lambda distribution with tail length parameter = ALAMBA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a11 | Usage: CALL LAMCDF(X,ALAMBA,CDF) | On-line doc: CALL GAMSDOC LAMCDF (or @PRT DATAPAC*DOC.LAMCDF) | Access: LIB NBS*DAPAC
- LAMPDF** Computes the probability density function value for the (Tukey) lambda distribution with tail length parameter ALAMBA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a11 | Usage: CALL LAMPDF(X,ALAMBA,PDF) | On-line doc: CALL GAMSDOC LAMPDF (or @PRT DATAPAC*DOC.LAMPDF) | Access: LIB NBS*DAPAC
- LAMPLT** Generates a (Tukey) lambda distribution probability plot with tail length parameter ALAMBA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c41 | Usage: CALL LAMPLT(X,N,ALAMBA) | On-line doc: CALL GAMSDOC LAMPLT (or @PRT DATAPAC*DOC.LAMPLT) | Access: LIB NBS*DAPAC
- LAMPPF** Computes the percent point function value for the (Tukey) lambda distribution with tail length parameter ALAMBA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a21 | Usage: CALL LAMPPF(P,ALAMBA,PPF) | On-line doc: CALL GAMSDOC LAMPPF (or @PRT DATAPAC*DOC.LAMPPF) | Access: LIB NBS*DAPAC
- LAMRAN** Generates a random sample of size N from the (Tukey) lambda distribution with tail length parameter ALAMBA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a12 | Usage: CALL LAMRAN(N,ALAMBA,ISTART,X) | On-line doc: CALL GAMSDOC LAMRAN (or @PRT DATAPAC*DOC.LAMRAN) | Access: LIB NBS*DAPAC
- LAMSF** Computes the sparsity function value for the (Tukey) lambda distribution with tail length parameter ALAMBA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a21 | Usage: CALL LAMSF(P,ALAMBA,SF) | On-line doc: CALL GAMSDOC LAMSF (or @PRT DATAPAC*DOC.LAMSF) | Access: LIB NBS*DAPAC
- LEAVE** Restores prior error recovery mode and reset the stack for PORT library programs. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): R3c | Usage: CALL LEAVE | On-line doc: CALL GAMSDOC LEAVE (or @PRT PORT*DOC.LEAVE) | Access: LIB NBS*PORT
- LEQ1PB** Linear equation solution - positive definite symmetric band matrix - band symmetric storage mode - space economizer solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b2 | Usage: CALL LEQ1PB (A,N,NC,IA,B,IB,M,IDGT,D1,D2,IER) | On-line doc: CALL GAMSDOC LEQ1PB (or @PRT IMSL*DOC.LEQ1PB) | Access: LIB NBS*IMSL
- LEQ1S** Linear equation solution - indefinite matrix - symmetric storage mode - space economizer solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b1a | Usage: CALL LEQ1S (A,N,B,M,IB,IJOB,ICHNG,DET,IER) | On-line doc: CALL GAMSDOC LEQ1S (or @PRT IMSL*DOC.LEQ1S) | Access: LIB NBS*IMSL
- LEQ2C** Linear equation solution - complex matrix high accuracy solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2c1 | Usage: CALL LEQ2C (A,N,IA,B,M,IB,IJOB,WA,WK,IER) | On-line doc: CALL GAMSDOC LEQ2C (or @PRT IMSL*DOC.LEQ2C) | Access: LIB NBS*IMSL
- LEQ2PB** Linear equation solution - positive definite band symmetric matrix - band symmetric storage mode - high accuracy solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b2 | Usage: CALL LEQ2PB (A,N,NC,IA,B,IB,M,IDGT,D1,D2,WK,IER) | On-line doc: CALL GAMSDOC LEQ2PB (or @PRT IMSL*DOC.LEQ2PB) | Access: LIB NBS*IMSL
- LEQ2S** Linear equation solution - indefinite matrix - symmetric storage mode - high accuracy solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b1a | Usage: CALL LEQ2S (A,N,B,M,IB,IJOB,ICHNG,DET,IER) | On-line doc: CALL GAMSDOC LEQ2S (or @PRT IMSL*DOC.LEQ2S) | Access: LIB NBS*IMSL
- LEQIF** Linear equation solution - full matrices (virtual memory version). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2a1 | Usage: CALL LEQIF(A,IA,N,MA,B,IB,M,IJOB,WK,IER) | On-line doc: CALL GAMSDOC LEQIF (or @PRT IMSL*DOC.LEQIF) | Access: LIB NBS*IMSL
- LEQOF** Linear equation solution - full matrices (out-of-core version). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2a1 | Usage: CALL LEQOF(IUNITA,IUNITW,N,MA,B,IB,M,IJOB,WK,IER) | On-line doc: CALL GAMSDOC LEQOF (or

@PRT IMSL*DOC.LEQOF) | Access: LIB NBS*IMSL

LEQT1B Linear equation solution - band storage mode - space economizer solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2a2 | Usage: CALL LEQT1B (A,N,NLC,NUC,IA,B,M,IB,IJOB,XL,IER) | On-line doc: CALL GAMSDOC LEQT1B (or @PRT IMSL*DOC.LEQT1B) | Access: LIB NBS*IMSL

LEQT1C Matrix decomposition, linear equation solution - space economizer solution complex matrices. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2c1 | Usage: CALL LEQT1C (A,N,IA,B,M,IB,IJOB,WA,IER) | On-line doc: CALL GAMSDOC LEQT1C (or @PRT IMSL*DOC.LEQT1C) | Access: LIB NBS*IMSL

LEQT1F Linear equation solution - full storage mode - space economizer solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2a1 | Usage: CALL LEQT1F (A,M,N,IA,B,IDGT,WKAREA,IER) | On-line doc: CALL GAMSDOC LEQT1F (or @PRT IMSL*DOC.LEQT1F) | Access: LIB NBS*IMSL

LEQT1P Linear equation solution - positive definite matrix - symmetric storage mode - space economizer solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b1b | Usage: CALL LEQT1P (A,M,N,B,IB,IDGT,D1,D2,IER) | On-line doc: CALL GAMSDOC LEQT1P (or @PRT IMSL*DOC.LEQT1P) | Access: LIB NBS*IMSL

LEQT2B Linear equation solution - band storage mode - high accuracy solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2a2 | Usage: CALL LEQT2B (A,N,NLC,NUC,IA,B,M,IB,IJOB,U,IU,LX,IER) | On-line doc: CALL GAMSDOC LEQT2B (or @PRT IMSL*DOC.LEQT2B) | Access: LIB NBS*IMSL

LEQT2F Linear equation solution - full storage mode - high accuracy solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2a1 | Usage: CALL LEQT2F (A,M,N,IA,B,IDGT,WKAREA,IER) | On-line doc: CALL GAMSDOC LEQT2F (or @PRT IMSL*DOC.LEQT2F) | Access: LIB NBS*IMSL

LEQT2P Linear equations solution - positive definite matrix - symmetric storage mode - high accuracy solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b1b | Usage: CALL LEQT2P (A,M,N,IB,B,IDGT,D1,D2,WKAREA,IER) | On-line doc: CALL GAMSDOC LEQT2P (or @PRT IMSL*DOC.LEQT2P) | Access: LIB NBS*IMSL

LGINF Generalized inverse of a real matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D0 | Usage: CALL LGINF (A,IA,M,N,TOL,AINV,IAINV,S,WK,IER) | On-line doc: CALL GAMSDOC LGINF (or @PRT IMSL*DOC.LGINF) | Access: LIB NBS*IMSL

LGNCDF Computes the cumulative distribution function value for the lognormal distribution with mean \sqrt{e} . | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a11 | Usage: CALL LGNCDF (X,CDF) | On-line doc: CALL GAMSDOC LGNCDF (or @PRT DATAPAC*DOC.LGNCDF) | Access: LIB NBS*DAPAC

LGNPLT Generates a lognormal probability plot with mean $= \sqrt{e}$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c4I | Usage: CALL LGNPLT (X,N) | On-line doc: CALL GAMSDOC LGNPLT (or @PRT DATAPAC*DOC.LGNPLT) | Access: LIB NBS*DAPAC

LGNPPF Computes the percent point function value for the lognormal distribution with mean $= \sqrt{e}$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2I | Usage: CALL LGNPPF (P,PPF) | On-line doc: CALL GAMSDOC LGNPPF (or @PRT DATAPAC*DOC.LGNPPF) | Access: LIB NBS*DAPAC

LGNRAN Generates a random sample of size N from the lognormal distribution with mean $= \sqrt{e}$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L8a12 | Usage: CALL LGNRAN (N,ISTART,X) | On-line doc: CALL GAMSDOC LGNRAN (or @PRT DATAPAC*DOC.LGNRAN) | Access: LIB NBS*DAPAC

LIN1PB Inversion of a matrix - positive definite band symmetric matrix - band symmetric storage mode - space economizer solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b2 | Usage: CALL LIN1PB (A,N,NC,IA,AINV,IDGT,D1,D2,IER) | On-line doc: CALL GAMSDOC LIN1PB (or @PRT IMSL*DOC.LIN1PB) | Access: LIB NBS*IMSL

LIN2PB Inversion of matrix - positive definite band symmetric matrix - band symmetric storage mode - high accuracy solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b2 | Usage: CALL LIN2PB (A,N,NC,IA,AINV,IDGT,D1,D2,WK,IER) | On-line doc: CALL GAMSDOC LIN2PB (or @PRT IMSL*DOC.LIN2PB) | Access: LIB NBS*IMSL

LINEQ Solves a real system of linear equations, $AX=B$, where B is allowed to be a matrix or a vector. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DLINEQ. | Class(es): D2a1 | Usage: CALL LINEQ (N, A, B, NB, X) | On-line doc: CALL GAMSDOC LINEQ (or @PRT PORT*DOC.LINEQ) | Access: LIB NBS*PORT

LINV1F Inversion of a matrix - full storage mode space economizer solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2a1 | Usage: CALL LINV1F (A,N,IA,AINV,IDGT,WKAREA,IER) | On-line doc: CALL GAMSDOC LINV1F (or @PRT IMSL*DOC.LINV1F) | Access: LIB NBS*IMSL

LINV1P Inversion of matrix - positive definite symmetric storage mode - space economizer solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b1b | Usage: CALL LINV1P (A,N,AINV,IDGT,D1,D2,IER) | On-line doc: CALL GAMSDOC LINV1P (or @PRT IMSL*DOC.LINV1P) | Access: LIB NBS*IMSL

- LINV2F** Inversion of a matrix - full storage mode high accuracy solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2a1 | Usage: CALL LINV2F (A,N,IA,AINV,IDGT,WKAREA,IER) | On-line doc: CALL GAMSDOC LINV2F (or @PRT IMSL*DOC.LINV2F) | Access: LIB NBS*IMSL
- LINV2P** Inversion of a matrix - positive definite symmetric storage mode - high accuracy solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b1b | Usage: CALL LINV2P (A,N,AINV,IDGT,D1,D2,WKAREA,IER) | On-line doc: CALL GAMSDOC LINV2P (or @PRT IMSL*DOC.LINV2P) | Access: LIB NBS*IMSL
- LINV3F** In place inverse, equation solution, and/or determinant evaluation - full storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2a1 D3a1 | Usage: CALL LINV3F (A,B,IJOB,N,IA,D1,D2,WKAREA,IER) | On-line doc: CALL GAMSDOC LINV3F (or @PRT IMSL*DOC.LINV3F) | Access: LIB NBS*IMSL
- LINV3P** In place inverse, equation solution, positive definite matrix - symmetric storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b1b | Usage: CALL LINV3P (A,B,IJOB,N,IER) | On-line doc: CALL GAMSDOC LINV3P (or @PRT IMSL*DOC.LINV3P) | Access: LIB NBS*IMSL
- LLBQF** Solution of linear least squares problem high accuracy solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D9 | Usage: CALL LLBQF (A,IA,M,N,B,IB,NB,IND,C,X,IX,IWK,WK,IER) | On-line doc: CALL GAMSDOC LLBQF (or @PRT IMSL*DOC.LLBQF) | Access: LIB NBS*IMSL
- LLSIA** Computes least squares solution to $AX=B$ with A an m by n matrix with m.ge.n Flexible version of SGLSS. | Portable single precision Fortran subprogram in SGLSS sublibrary of CMLIB library. | Class(es): D9 | Usage: CALL LLSIA(A,MDA,M,N,B,MDB,NB,RE,AE,KEY,MODE,NP, KRANK,KSURE, RNORM,W,LW,IWORK,LIW,INFO) | On-line doc: CALL GAMSDOC LLSIA (or @PRT CMLIB*DOC.LLSIA/SGLSS) | Access: LIB NBS*CMLIB
- LLSQF** Solution of a linear least squares problem. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D9 | Usage: CALL LLSQF (A,IA,M,N,B,TOL,KBASIS,X,H,IP,IER) | On-line doc: CALL GAMSDOC LLSQF (or @PRT IMSL*DOC.LLSQF) | Access: LIB NBS*IMSL
- LOC** Computes 4 estimates (midrange, mean, midmean, and median) of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1 | Usage: CALL LOC(X,N) | On-line doc: CALL GAMSDOC LOC (or @PRT DATAPAC*DOC.LOC) | Access: LIB NBS*DAPAC
- LOGCDF** Computes the cumulative distribution function value for the logistic distribution with mean = 0 and standard deviation = $\pi/\sqrt{3}$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a11 | Usage: CALL LOGCDF(X,CDF) | On-line doc: CALL GAMSDOC LOGCDF (or @PRT DATAPAC*DOC.LOGCDF) | Access: LIB NBS*DAPAC
- LOGPDF** Computes the probability density function value for the logistic distribution with mean = 0 and standard deviation = $\pi/\sqrt{3}$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a11 | Usage: CALL LOGPDF(X,PDF) | On-line doc: CALL GAMSDOC LOGPDF (or @PRT DATAPAC*DOC.LOGPDF) | Access: LIB NBS*DAPAC
- LOGPLT** Generates a logistic probability plot with mean = 0 and standard deviation = $\pi/\sqrt{3}$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c4I | Usage: CALL LOGPLT(X,N) | On-line doc: CALL GAMSDOC LOGPLT (or @PRT DATAPAC*DOC.LOGPLT) | Access: LIB NBS*DAPAC
- LOGPPF** Computes the percent point function value for the logistic distribution with mean = 0 and standard deviation = $\pi/\sqrt{3}$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a21 | Usage: CALL LOGPPF(P,PPF) | On-line doc: CALL GAMSDOC LOGPPF (or @PRT DATAPAC*DOC.LOGPPF) | Access: LIB NBS*DAPAC
- LOGRAN** Generates a random sample of size N from the logistic distribution with mean = 0 and standard deviation = $\pi/\sqrt{3}$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a12 | Usage: CALL LOGRAN(N,ISTART,X) | On-line doc: CALL GAMSDOC LOGRAN (or @PRT DATAPAC*DOC.LOGRAN) | Access: LIB NBS*DAPAC
- LOGSF** Computes the sparsity function value for the logistic distribution with mean = 0 and standard deviation = $\pi/\sqrt{3}$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a21 | Usage: CALL LOGSF(P,SF) | On-line doc: CALL GAMSDOC LOGSF (or @PRT DATAPAC*DOC.LOGSF) | Access: LIB NBS*DAPAC
- LOTPS** Passes smooth function thru points $(X_i(K), Y_i(K), F_i(K), i=1..NPI)$ and returns an array of interpolated values on user specified grid. | Portable single precision Fortran subprogram in LOTPS sublibrary of CMLIB library. | Class(es): E2b | Usage: CALL LOTPS(MODE,NPPR,NPI,XI,YI,FI,NXO,XO,NYO,YO,IWK,NIWK,IJWKU, WK, NWK,FO,KER) | On-line doc: CALL GAMSDOC LOTPS (or @PRT CMLIB*DOC.LOTPS/LOTPTS) | Tests: CMLIB*TEST-SOURCE.\$F/LOTPTS, CMLIB*TEST-SOURCE.\$Q/LOTPTS | Access: LIB NBS*CMLIB
- LPDP** Solves least projected distance problem. | Portable single precision Fortran subprogram in FC sublibrary of CMLIB library. | Class(es): K1a2a | Usage: CALL LPDP(A,MDA,M,N1,N2,PRGOPT,X,WNORM,MODE,WS,IS) | On-line doc: CALL GAMSDOC LPDP (or @PRT CMLIB*DOC.LPDP/FC) | Tests: CMLIB*TEST-SOURCE.\$F/FC | Access: LIB NBS*CMLIB
- LPLOT** Prints a letter plot with symbols corresponding to numerical "tag" values. Scale specification is optional. | Command in MINITAB Proprietary interactive system. | Class(es): L3c2 Q1 | Usage: LPLOT C [from K to K] vs C [from K to K] using tags in C | On-line

doc: HELP LPLOT (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

- LSEI** Solves linearly constrained least squares problem with equality and inequality constraints. Covariance matrix opt output. | Portable single precision Fortran subprogram in FC sublibrary of CMLIB library. | Class(es): K1a2a D9 | Usage: CALL LSEI(W,MDW,ME,MA,MG,N,PRGOPT,X,RNORME,RNORMML,MODE,WS,IP) | On-line doc: CALL GAMSDOC LSEI (or @PRT CMLIB*DOC.LSEI/FC) | Tests: CMLIB*TEST-SOURCE.\$F/FC | Access: LIB NBS*CMLIB
- LSTSQ** Finds the least squares solution of a system of linear equations, $AX=B$. B may be a matrix. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DLSTSQ. | Class(es): D9 | Usage: CALL LSTSQ (MDIM,NDIM,M,N,A,B,NB,X) | On-line doc: CALL GAMSDOC LSTSQ (or @PRT PORT*DOC.LSTSQ) | Access: LIB NBS*PORT
- LSVDB** Singular value decomposition of a bidiagonal matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D6 | Usage: CALL LSVDB (D,E,N,V,IV,NRV,C,IC,NCC,IER) | On-line doc: CALL GAMSDOC LSVDB (or @PRT IMSL*DOC.LSVDB) | Access: LIB NBS*IMSL
- LSVDF** Singular value decomposition of a real matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D6 | Usage: CALL LSVDF (A,IA,M,N,B,IB,NB,S,WK,IER) | On-line doc: CALL GAMSDOC LSVDF (or @PRT IMSL*DOC.LSVDF) | Access: LIB NBS*IMSL
- LUDAPB** Decomposition of a positive definite band symmetric matrix - band symmetric storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b2 | Usage: CALL LUDAPB (A,N,NC,IA,UL,IU,D1,D2,IER) | On-line doc: CALL GAMSDOC LUDAPB (or @PRT IMSL*DOC.LUDAPB) | Access: LIB NBS*IMSL
- LUDATF** L-U decomposition by the Crout algorithm with optional accuracy test. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2a1 | Usage: CALL LUDATF (A,LU,N,IA,IDGT,D1,D2,IPVT,EQUIL,WA,IER) | On-line doc: CALL GAMSDOC LUDATF (or @PRT IMSL*DOC.LUDATF) | Access: LIB NBS*IMSL
- LUDECP** Decomposition of a positive definite matrix symmetric storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b1b | Usage: CALL LUDECP (A,UL,N,D1,D2,IER) | On-line doc: CALL GAMSDOC LUDECP (or @PRT IMSL*DOC.LUDECP) | Access: LIB NBS*IMSL
- LUELMF** Elimination part of solution of $Ax=b$ (full storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2a1 | Usage: CALL LUELMF (A,B,IPVT,N,IA,X) | On-line doc: CALL GAMSDOC LUELMF (or @PRT IMSL*DOC.LUELMF) | Access: LIB NBS*IMSL | See also: LUDATF
- LUELMP** Elimination part of the solution of $Ax=b$ positive definite matrix - symmetric storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b1b | Usage: CALL LUELMP (A,B,N,X) | On-line doc: CALL GAMSDOC LUELMP (or @PRT IMSL*DOC.LUELMP) | Access: LIB NBS*IMSL | See also: LUDECP
- LUELPB** Elimination part of solution of $Ax=b$ positive definite band symmetric matrix band symmetric storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b2 | Usage: CALL LUELPB (UL,B,N,NC,IA,X) | On-line doc: CALL GAMSDOC LUELPB (or @PRT IMSL*DOC.LUELPB) | Access: LIB NBS*IMSL | See also: LUDAPB
- LUMB** Given a basic mesh, this subdivides each interval uniformly for B-spline use. Multiplicities are allowed. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DLUMB. | Class(es): E3 K6 | Usage: CALL LUMB (XB,NXB,N,K,X,NX) | On-line doc: CALL GAMSDOC LUMB (or @PRT PORT*DOC.LUMB) | Access: LIB NBS*PORT
- LUMD** Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is IDUMD. | Class(es): E3 K6 | Usage: CALL LUMD (XB,NXB,N,X,NX) | On-line doc: CALL GAMSDOC LUMD (or @PRT PORT*DOC.LUMD) | Access: LIB NBS*PORT
- LUREFF** Refinement of solution to linear equations full storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2a1 | Usage: CALL LUREFF (A,B,UL,IPVT,N,IA,X,IDGT,RES,DX,IER) | On-line doc: CALL GAMSDOC LUREFF (or @PRT IMSL*DOC.LUREFF) | Access: LIB NBS*IMSL | See also: LUDATF LUELMF
- LUREFP** Refinement of solution to linear equations positive definite matrix - symmetric storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b1b | Usage: CALL LUREFP (A,B,UL,N,X,IDGT,RES,IER) | On-line doc: CALL GAMSDOC LUREFP (or @PRT IMSL*DOC.LUREFP) | Access: LIB NBS*IMSL | See also: LUDECP LUELMP
- LUREPB** Refinement of solution to linear equations positive definite band symmetric matrix band symmetric storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b2 | Usage: CALL LUREPB (A,N,NC,IA,UL,IU,B,X,IDGT,RES,IER) | On-line doc: CALL GAMSDOC LUREPB (or @PRT IMSL*DOC.LUREPB) | Access: LIB NBS*IMSL | See also: LUDAPB
- LVALS** Prints letter-value display - median, hinges, eighths, etc., and optionally saves results. | Command in MINITAB Proprietary interactive system. Class(es): L3d | Usage: LVALS display of C [put letter values into C [mids into C [spreads into C]]] | On-line doc: HELP LVALS (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

M

- M01AAE** Passively sort a real vector into ascending order. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AAF. | Class(es): N6a1b1 | Usage: CALL M01AAE (A, M, N, IP, IST, IFAIL) | On-line doc: CALL GAMSDOC M01AAE (or @PRT NAG*DOC.M01AAE) | Access: LIB NBS*NAG
- M01AAF** Passively sort a real vector into ascending order. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AAE. | Class(es): N6a1b2 | Usage: CALL M01AAF (A, M, N, IP, IST, IFAIL) | On-line doc: CALL GAMSDOC M01AAF (or @PRT NAG*DOC.M01AAF) | Access: LIB NBS*NAG
- M01ABE** Passively sort a real vector into descending order. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01ABF. | Class(es): N6a1b1 | Usage: CALL M01ABE (A, M, N, IP, IST, IFAIL) | On-line doc: CALL GAMSDOC M01ABE (or @PRT NAG*DOC.M01ABE) | Access: LIB NBS*NAG
- M01ABF** Passively sort a real vector into descending order. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01ABE. | Class(es): N6a1b2 | Usage: CALL M01ABF (A, M, N, IP, IST, IFAIL) | On-line doc: CALL GAMSDOC M01ABF (or @PRT NAG*DOC.M01ABF) | Access: LIB NBS*NAG
- M01ACE** Passively sort an integer vector into ascending order. (Identical to M01ACF.). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01ACF. | Class(es): N6a1a | Usage: CALL M01ACE (IA, M, N, IP, IST, IFAIL) | On-line doc: CALL GAMSDOC M01ACE (or @PRT NAG*DOC.M01ACE) | Access: LIB NBS*NAG
- M01ACF** Passively sort an integer vector into ascending order. (Identical to M01ACE.). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01ACE. | Class(es): N6a1a | Usage: CALL M01ACF (IA, M, N, IP, IST, IFAIL) | On-line doc: CALL GAMSDOC M01ACF (or @PRT NAG*DOC.M01ACF) | Access: LIB NBS*NAG
- M01ADE** Passively sort an integer vector into descending order. (Identical to M01ADF.). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01ADF. | Class(es): N6a1a | Usage: CALL M01ADE (IA, M, N, IP, IST, IFAIL) | On-line doc: CALL GAMSDOC M01ADE (or @PRT NAG*DOC.M01ADE) | Access: LIB NBS*NAG
- M01ADF** Passively sort an integer vector into descending order. (Identical to M01ADE.). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01ADE. | Class(es): N6a1a | Usage: CALL M01ADF (IA, M, N, IP, IST, IFAIL) | On-line doc: CALL GAMSDOC M01ADF (or @PRT NAG*DOC.M01ADF) | Access: LIB NBS*NAG
- M01AEE** Actively sort the rows of a real matrix into ascending order of an index column. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AEF. | Class(es): N6a2b1 | Usage: CALL M01AEE (A, NR, NC, IC, T, TT, IFAIL) | On-line doc: CALL GAMSDOC M01AEE (or @PRT NAG*DOC.M01AEE) | Access: LIB NBS*NAG
- M01AEF** Actively sort the rows of a real matrix into ascending order of an index column. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AEE. | Class(es): N6a2b2 | Usage: CALL M01AEF (A, NR, NC, IC, T, TT, IFAIL) | On-line doc: CALL GAMSDOC M01AEF (or @PRT NAG*DOC.M01AEF) | Access: LIB NBS*NAG
- M01AFE** Actively sort the rows of a real matrix into descending order of an index column. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AFF. | Class(es): N6a2b1 | Usage: CALL M01AFE (A, NR, NC, IC, T, TT, IFAIL) | On-line doc: CALL GAMSDOC M01AFE (or @PRT NAG*DOC.M01AFE) | Access: LIB NBS*NAG
- M01AFF** Actively sort the rows of a real matrix into descending order of an index column. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AFE. | Class(es): N6a2b2 | Usage: CALL M01AFF (A, NR, NC, IC, T, TT, IFAIL) | On-line doc: CALL GAMSDOC M01AFF (or @PRT NAG*DOC.M01AFF) | Access: LIB NBS*NAG
- M01AGE** Actively sort the rows of an integer matrix into ascending order of an index column. (Identical to M01AGF.). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AGF. | Class(es): N6a2a | Usage: CALL M01AGE (IA, NR, NC, IC, K, L, IFAIL) | On-line doc: CALL GAMSDOC M01AGE (or @PRT NAG*DOC.M01AGE) | Access: LIB NBS*NAG
- M01AGF** Actively sort the rows of an integer matrix into ascending order of an index column. (Identical to M01AGE.). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AGE. | Class(es): N6a2a | Usage: CALL M01AGF (IA, NR, NC, IC, K, L, IFAIL) | On-line doc: CALL GAMSDOC M01AGF (or @PRT NAG*DOC.M01AGF) | Access: LIB NBS*NAG
- M01AHE** Actively sort the rows of an integer matrix into descending order of an index column. (Identical to M01AHF.). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AHF. | Class(es): N6a2a | Usage: CALL M01AHE (IA, NR, NC, IC, K, L, IFAIL) | On-line doc: CALL GAMSDOC M01AHE (or @PRT NAG*DOC.M01AHE) | Access: LIB NBS*NAG
- M01AHF** Actively sort the rows of an integer matrix into descending order of an index column. (Identical to M01AHE.). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AHE. | Class(es): N6a2a | Usage: CALL M01AHF (IA, NR, NC, IC, K, L, IFAIL) | On-line doc: CALL GAMSDOC M01AHF (or @PRT NAG*DOC.M01AHF) | Access: LIB NBS*NAG
- M01AJE** Actively sort a real vector into ascending order and provide an index to the original order. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AJF. | Class(es): N6a2b1 | Usage: CALL M01AJE (A, W, IND, INDW, N,

NW, IFAIL) | On-line doc: CALL GAMSDOC M01AJE (or @PRT NAG*DOC.M01AJE) | Access: LIB NBS*NAG

M01AJF Actively sort a real vector into ascending order and provide an index to the original order. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AJE. | Class(es): N6a2b2 | Usage: CALL M01AJF (A, W, IND, INDW, N, NW, IFAIL) | On-line doc: CALL GAMSDOC M01AJF (or @PRT NAG*DOC.M01AJF) | Access: LIB NBS*NAG

M01AKE Actively sort a real vector into descending order and provide an index to the original order. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AKF. | Class(es): N6a2b1 | Usage: CALL M01AKE (A, W, IND, INDW, N, NW, IFAIL) | On-line doc: CALL GAMSDOC M01AKE (or @PRT NAG*DOC.M01AKE) | Access: LIB NBS*NAG

M01AKF Actively sort a real vector into descending order and provide an index to the original order. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AKE. | Class(es): N6a2b2 | Usage: CALL M01AKF (A, W, IND, INDW, N, NW, IFAIL) | On-line doc: CALL GAMSDOC M01AKF (or @PRT NAG*DOC.M01AKF) | Access: LIB NBS*NAG

M01ALE Actively sort an integer vector into ascending order of an index column and provide an index to the original order. (Identical to M01ALF.). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01ALF. | Class(es): N6a2a | Usage: CALL M01ALE (IA, IW, IND, INDW, N, NW, IFAIL) | On-line doc: CALL GAMSDOC M01ALE (or @PRT NAG*DOC.M01ALE) | Access: LIB NBS*NAG

M01ALF Actively sort an integer vector into ascending order of an index column and provide an index to the original order. (Identical to M01ALE.). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01ALE. | Class(es): N6a2a | Usage: CALL M01ALF (IA, IW, IND, INDW, N, NW, IFAIL) | On-line doc: CALL GAMSDOC M01ALF (or @PRT NAG*DOC.M01ALF) | Access: LIB NBS*NAG

M01AME Actively sort an integer vector into descending order of an index column and provide an index to the original order. (Identical to M01AMF.). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AMF. | Class(es): N6a2a | Usage: CALL M01AME (IA, IW, IND, INDW, N, NW, IFAIL) | On-line doc: CALL GAMSDOC M01AME (or @PRT NAG*DOC.M01AME) | Access: LIB NBS*NAG

M01AMF Actively sort an integer vector into descending order of an index column and provide an index to the original order. (Identical to M01AME.). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AME. | Class(es): N6a2a | Usage: CALL M01AMF (IA, IW, IND, INDW, N, NW, IFAIL) | On-line doc: CALL GAMSDOC M01AMF (or @PRT NAG*DOC.M01AMF) | Access: LIB NBS*NAG

M01ANE Actively sort a real vector into ascending order (Singleton's implementation of Quicksort). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01ANF. | Class(es): N6a2b1 | Usage: CALL M01ANE (A, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01ANE (or @PRT NAG*DOC.M01ANE) | Access: LIB NBS*NAG

M01ANF Actively sort a real vector into ascending order (Singleton's implementation of Quicksort). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01ANE. | Class(es): N6a2b2 | Usage: CALL M01ANF (A, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01ANF (or @PRT NAG*DOC.M01ANF) | Access: LIB NBS*NAG

M01APE Actively sort a real vector into descending order (Singleton's implementation of Quicksort). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01APF. | Class(es): N6a2b1 | Usage: CALL M01APE (A, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01APE (or @PRT NAG*DOC.M01APE) | Access: LIB NBS*NAG

M01APF Actively sort a real vector into descending order (Singleton's implementation of Quicksort). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01APE. | Class(es): N6a2b2 | Usage: CALL M01APF (A, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01APF (or @PRT NAG*DOC.M01APF) | Access: LIB NBS*NAG

M01AQE Actively sort an integer vector into ascending order (Singleton's implementation of Quicksort). (Identical to M01AQF.). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AQF. | Class(es): N6a2a | Usage: CALL M01AQE (IA, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01AQE (or @PRT NAG*DOC.M01AQE) | Access: LIB NBS*NAG

M01AQF Actively sort an integer vector into ascending order (Singleton's implementation of Quicksort). (Identical to M01AQE.). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AQE. | Class(es): N6a2a | Usage: CALL M01AQF (IA, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01AQF (or @PRT NAG*DOC.M01AQF) | Access: LIB NBS*NAG

M01ARE Actively sort an integer vector into descending order (Singleton's implementation of Quicksort). (Identical to M01ARF.). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01ARF. | Class(es): N6a2a | Usage: CALL M01ARE (IA, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01ARE (or @PRT NAG*DOC.M01ARE) | Access: LIB NBS*NAG

M01ARF Actively sort an integer vector into descending order (Singleton's implementation of Quicksort). (Identical to M01ARE.). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01ARE. | Class(es): N6a2a | Usage: CALL M01ARF (IA, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01ARF (or @PRT NAG*DOC.M01ARF) | Access: LIB NBS*NAG

M01BAE Actively sort a character vector into reverse alphanumeric order (Singleton's implementation of Quicksort). (Identical to M01BAF.). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01BAF. | Class(es): N6a2c | Usage: CALL M01BAE (IA, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01BAE (or @PRT NAG*DOC.M01BAE) | Access: LIB NBS*NAG

M01BAF Actively sort a character vector into reverse alphanumeric order (Singleton's implementation of Quicksort). (Identical to M01BAE.).

- | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01BAE. | Class(es): N6a2c | Usage: CALL M01BAF (IA, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01BAF (or @PRT NAG*DOC.M01BAF) | Access: LIB NBS*NAG
- M01BBE** Actively sort a character vector into alphanumeric order (Singleton's implementation of Quicksort). (Identical to M01BBF.). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01BBF. | Class(es): N6a2c | Usage: CALL M01BBE (IA, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01BBE (or @PRT NAG*DOC.M01BBE) | Access: LIB NBS*NAG
- M01BBF** Actively sort a character vector in alphanumeric order (Singleton's implementation of Quicksort). (Identical to M01BBE.). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01BBE. | Class(es): N6a2c | Usage: CALL M01BBF (IA, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01BBF (or @PRT NAG*DOC.M01BBF) | Access: LIB NBS*NAG
- M01BCE** Actively sort the columns of a character matrix into reverse alphanumeric order of an index column. (Identical to M01BCF.). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01BCF. | Class(es): N6a2c | Usage: CALL M01BCE (IA, NR, NC, L1, L2, LC, IUC, IT, ITT, IFAIL) | On-line doc: CALL GAMSDOC M01BCE (or @PRT NAG*DOC.M01BCE) | Access: LIB NBS*NAG
- M01BCF** Actively sort the columns of a character matrix into reverse alphanumeric order of an index column. (Identical to M01BCE.). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01BCE. | Class(es): N6a2c | Usage: CALL M01BCF (IA, NR, NC, L1, L2, LC, IUC, IT, ITT, IFAIL) | On-line doc: CALL GAMSDOC M01BCF (or @PRT NAG*DOC.M01BCF) | Access: LIB NBS*NAG
- M01BDE** Actively sort the columns of a character matrix into alphanumeric order of an index column. (Identical to M01BDF.). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01BDF. | Class(es): N6a2c | Usage: CALL M01BDE (IA, NR, NC, L1, L2, LC, IUC, IT, ITT, IFAIL) | On-line doc: CALL GAMSDOC M01BDE (or @PRT NAG*DOC.M01BDE) | Access: LIB NBS*NAG
- M01BDF** Actively sort the columns of a character matrix into alphanumeric order of an index column. (Identical to M01BDE.). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01BDE. | Class(es): N6a2c | Usage: CALL M01BDF (IA, NR, NC, L1, L2, LC, IUC, IT, ITT, IFAIL) | On-line doc: CALL GAMSDOC M01BDF (or @PRT NAG*DOC.M01BDF) | Access: LIB NBS*NAG
- MANN-WHITNEY** Performs one- or two-sided two-sample rank test (a.k.a. Wilcoxon rank test) for the difference between two population medians, and calculates the corresponding point and confidence interval estimates. | Command in MINITAB Proprietary interactive system. Class(es): L4b1b | Usage: MANN-Whitney [alternative K], [percent confidence K] for data in C and C | On-line doc: HELP MANN-WHITNEY (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- MATLAB** An interactive system for defining and manipulating matrices. It includes solving linear systems, linear least squares, eigenvalue and eigenvector calculation, QR decomposition, singular value decomposition and inverses, as well as other interesting and useful features. | Portable mixed precision Fortran subprogram in MATLAB library. | Class(es): D | On-line doc: @PRT MATLAB*DOC.SUMMARY | Access: @XQT NBS*MATLAB.MATLAB (or XQT NBS*MATLAB.MATLAB i
- MAX** Computes the sample maximum of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): D1a2 | Usage: CALL MAX(X,N,IWRITE,XMAX) | On-line doc: CALL GAMSDOC MAX (or @PRT DATAPAC*DOC.MAX) | Access: LIB NBS*DAPAC
- MDBETA** Beta probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1b | Usage: CALL MDBETA (X,A,B,P,IER) | On-line doc: CALL GAMSDOC MDBETA (or @PRT IMSL*DOC.MDBETA) | Access: LIB NBS*IMSL
- MDBETI** Inverse beta probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a2b | Usage: CALL MDBETI (P,A,B,X,IER) | On-line doc: CALL GAMSDOC MDBETI (or @PRT IMSL*DOC.MDBETI) | Access: LIB NBS*IMSL
- MDBIN** Binomial probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1b | Usage: CALL MDBIN (K,N,P,PS,PK,IER) | On-line doc: CALL GAMSDOC MDBIN (or @PRT IMSL*DOC.MDBIN) | Access: LIB NBS*IMSL
- MDBNOR** Bivariate normal probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5b1n | Usage: CALL MDBNOR (X,Y,RHO,P,IER) | On-line doc: CALL GAMSDOC MDBNOR (or @PRT IMSL*DOC.MDBNOR) | Access: LIB NBS*IMSL
- MDCH** Chi-squared probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1c | Usage: CALL MDCH (CS,DF,P,IER) | On-line doc: CALL GAMSDOC MDCH (or @PRT IMSL*DOC.MDCH) | Access: LIB NBS*IMSL
- MDCHI** Inverse chi-squared probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a2c | Usage: CALL MDCHI (P,DF,X,IER) | On-line doc: CALL GAMSDOC MDCHI (or @PRT IMSL*DOC.MDCHI) | Access: LIB NBS*IMSL
- MDCHN** Non-central chi-squared probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1c | Usage: CALL MDCHN (CS,DF,PNONC,P,IER) | On-line doc: CALL GAMSDOC MDCHN (or @PRT IMSL*DOC.MDCHN) | Access: LIB NBS*IMSL
- MDFD** F probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1f | Usage:

CALL MDFD (F,N1,N2,P,IER) | On-line doc: CALL GAMSDOC MDFD (or @PRT IMSL*DOC.MDFD) | Access: LIB NBS*IMSL

MDFDRE F probability distribution function (integer or fractional degrees of freedom). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1f | Usage: CALL MDFDRE (X,DFN,DFD,P,IER) | On-line doc: CALL GAMSDOC MDFDRE (or @PRT IMSL*DOC.MDFDRE) | Access: LIB NBS*IMSL

MDFI Inverse F probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a2f | Usage: CALL MDFI (P,D1,D2,X,IER) | On-line doc: CALL GAMSDOC MDFI (or @PRT IMSL*DOC.MDFI) | Access: LIB NBS*IMSL

MDGAM Gamma probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1g | Usage: CALL MDGAM (X,P,PROB,IER) | On-line doc: CALL GAMSDOC MDGAM (or @PRT IMSL*DOC.MDGAM) | Access: LIB NBS*IMSL

MDGC General cumulative probability distribution function, given ordinates of the density. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1g | Usage: CALL MDGC (X,F,M,IOPT,B,C,IC,P,IER) | On-line doc: CALL GAMSDOC MDGC (or @PRT IMSL*DOC.MDGC) | Access: LIB NBS*IMSL

MDGCI Inverse of a general cumulative probability distribution function, given ordinates of the density. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a2g | Usage: CALL MDGCI (P,F,M,IOPT,B,C,IC,X,IER) | On-line doc: CALL GAMSDOC MDGCI (or @PRT IMSL*DOC.MDGCI) | Access: LIB NBS*IMSL

MDHYP Hypergeometric probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1h | Usage: CALL MDHYP (K,N,L,ND,PEQK,PLEK,IER) | On-line doc: CALL GAMSDOC MDHYP (or @PRT IMSL*DOC.MDHYP) | Access: LIB NBS*IMSL

MDNOR Normal or Gaussian probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1n C8a | Usage: CALL MDNOR (Y,P) | On-line doc: CALL GAMSDOC MDNOR (or @PRT IMSL*DOC.MDNOR) | Access: LIB NBS*IMSL

MDNRIS Inverse standard normal (Gaussian) probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a2n | Usage: CALL MDNRIS (P,Y,IER) | On-line doc: CALL GAMSDOC MDNRIS (or @PRT IMSL*DOC.MDNRIS) | Access: LIB NBS*IMSL

MDSMR Kolmogorov-Smirnov statistics asymptotic probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1k | Usage: CALL MDSMR (X,P1,P2) | On-line doc: CALL GAMSDOC MDSMR (or @PRT IMSL*DOC.MDSMR) | Access: LIB NBS*IMSL

MDSTI Inverse of a modification of Student's t probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a2t | Usage: CALL MDSTI (Q,F,X,IER) | On-line doc: CALL GAMSDOC MDSTI (or @PRT IMSL*DOC.MDSTI) | Access: LIB NBS*IMSL

MDTD Students t probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1t | Usage: CALL MDTD (TVAL,DF,Q,IER) | On-line doc: CALL GAMSDOC MDTD (or @PRT IMSL*DOC.MDTD) | Access: LIB NBS*IMSL

MDTN Non-central t probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1t | Usage: CALL MDTN (TVAL,IDF,D,P,IER) | On-line doc: CALL GAMSDOC MDTN (or @PRT IMSL*DOC.MDTN) | Access: LIB NBS*IMSL

MDTNF Integral related to calculation of noncentral t and bivariate normal probability distribution functions. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1t L5b1n | Usage: CALL MDTNF (Y,Z,EPS,T) | On-line doc: CALL GAMSDOC MDTNF (or @PRT IMSL*DOC.MDTNF) | Access: LIB NBS*IMSL

MDTPS Cumulative probability and, optionally, individual terms of the Poisson probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1p | Usage: CALL MDTPS (K,RLAM,IOPT,T,P) | On-line doc: CALL GAMSDOC MDTPS (or @PRT IMSL*DOC.MDTPS) | Access: LIB NBS*IMSL

MEAN Computes the sample mean of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1a | Usage: CALL MEAN(X,N,IWRITE,XMEAN) | On-line doc: CALL GAMSDOC MEAN (or @PRT DATAPAC*DOC.MEAN) | Access: LIB NBS*DAPAC

MEDIAN Computes the sample median of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1a | Usage: CALL MEDIAN(X,N,IWRITE,XMED) | On-line doc: CALL GAMSDOC MEDIAN (or @PRT DATAPAC*DOC.MEDIAN) | Access: LIB NBS*DAPAC

MERFCI Inverse complemented error function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): C8a L5a2e | Usage: CALL MERFCI (P,Y,IER) | On-line doc: CALL GAMSDOC MERFCI (or @PRT IMSL*DOC.MERFCI) | Access: LIB NBS*IMSL

MERFI Inverse error function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): C8a L5a2e | Usage: CALL MERFI (P,Y,IER) | On-line doc: CALL GAMSDOC MERFI (or @PRT IMSL*DOC.MERFI) | Access: LIB NBS*IMSL

MERRCZ Computes approximate values of $\exp(-z^2) \operatorname{erfc}(-iz)$ for complex z. | Proprietary double precision Fortran subprogram

in IMSL library. | Class(es): C8a | Usage: CALL MERRCZ(Z,W,IER) | On-line doc: CALL GAMSDOC MERRCZ (or @PRT IMSL*DOC.MERRCZ) | Access: LIB NBS*IMSL

MIDM Computes the sample midmean, i.e. the sample 25% (on each side) trimmed mean of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1a | Usage: CALL MIDM(X,N,IWRITE,XMIDM) | On-line doc: CALL GAMSDOC MIDM (or @PRT DATAPAC*DOC.MIDM) | Access: LIB NBS*DATAPAC

MIDR Computes the sample midrange of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1b | Usage: CALL MIDR(X,N,IWRITE,XMIDR) | On-line doc: CALL GAMSDOC MIDR (or @PRT DATAPAC*DOC.MIDR) | Access: LIB NBS*DATAPAC

MIN Computes the sample minimum of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): D1a2 | Usage: CALL MIN(X,N,IWRITE,XMIN) | On-line doc: CALL GAMSDOC MIN (or @PRT DATAPAC*DOC.MIN) | Access: LIB NBS*DATAPAC

MINFIT Compute Singular Value Decomposition of rectangular real matrix and solve related Linear Least Squares problem. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D9 D9 | Usage: CALL MINFIT(NM,M,N,A,W,IP,B,IERR,RV1) | On-line doc: CALL GAMSDOC MINFIT (or @PRT CMLIB*DOC.MINFIT/EISPACK) | Access: LIB NBS*CMLIB

MINITAB Minitab's vector summarization commands include COUNT, SUM, MEAN, MAX, MIN, MEDIAN, N, NMISS (number of missing values), STDEV, SSQ (sum of squares), DESCRIBE (N, MEAN, MEDIAN, STDEV, MAX, MIN, 5% trimmed mean, quartiles) for columns or rows (use prefix R, e.g., RMEAN) of data in the Minitab worksheet. | Command in MINITAB Proprietary interactive system. | Class(es): L1a1 L1a2 | On-line doc: HELP (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

MINITAB Minitab's vector and matrix commands include COPY (vectors, vectors to matrices, and conversely), DIAGONAL (create a diagonal matrix or extract the diagonal of a matrix), TRANSPOSE, INVERSE, and EIGEN (calculate eigenvalues and eigenvectors for a symmetric matrix). | Command in MINITAB Proprietary interactive system. | Class(es): D | On-line doc: HELP (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

MINITAB Minitab's subsetting commands include PICK, CHOOSE, and OMIT for selecting or deleting entries in a vector in a Minitab worksheet. | Command in MINITAB Proprietary interactive system. | Class(es): L2d | On-line doc: HELP (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

MINITAB Minitab's vector transformation commands include ADD, SUBTRACT, MULTIPLY, DIVIDE, RAISE, SIN, COS, TAN, ASIN, ACOS, ATAN, LOGE, LOGTEN, EXPONENTIAL, ANTILOG, ABSOLUTE VALUE, ROUND, SIGNS, sqrt, INDICATOR, RECODE, SUBSTITUTE, CONVERT, PARSUM, PARPRODUCT, and LET (to combine commands). | Command in MINITAB Proprietary interactive system. | Class(es): L2a | On-line doc: HELP (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

MINITAB Minitab's vector transformation commands include ADD, SUBTRACT, MULTIPLY, DIVIDE, RAISE, SIN, COS, TAN, ASIN, ACOS, ATAN, LOGE, LOGTEN, EXPONENTIAL, ANTILOG, ABSOLUTE VALUE, ROUND, SIGNS, SQRT, INDICATOR, RECODE, SUBSTITUTE, CONVERT, PARSUM, PARPRODUCT, and LET (to combine commands). | Command in MINITAB Proprietary interactive system. | Class(es): L2a | On-line doc: HELP (in Minitab) | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

MMBSI0 Modified Bessel function of the first kind of order zero. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C10b1 | Usage: D = MMBSI0 (IOPT,ARG,IER) | On-line doc: CALL GAMSDOC MMBSI0 (or @PRT IMSL*DOC.MMBSI0) | Access: LIB NBS*IMSL

MMBSI1 Modified Bessel function of the first kind of order one. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C10b1 | Usage: D = MMBSI1 (IOPT,ARG,IER) | On-line doc: CALL GAMSDOC MMBSI1 (or @PRT IMSL*DOC.MMBSI1) | Access: LIB NBS*IMSL

MMBSIN Computes approximate values of the modified Bessel function of the first kind of nonnegative integer order for real arguments. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C10b1 | Usage: CALL MMBSIN(ARG,N,B,IER) | On-line doc: CALL GAMSDOC MMBSIN (or @PRT IMSL*DOC.MMBSIN) | Access: LIB NBS*IMSL

MMBSIR Modified Bessel function of the first kind of nonnegative real order for real positive arguments with exponential scaling option. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C10b3 | Usage: CALL MMBSIR(ARG,ORDER,NB,IOPT,B,IER) | On-line doc: CALL GAMSDOC MMBSIR (or @PRT IMSL*DOC.MMBSIR) | Access: LIB NBS*IMSL

MMBSJ0 Bessel function of the first kind of order zero. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C10a1 | Usage: D = MMBSJ0 (ARG,IER) | On-line doc: CALL GAMSDOC MMBSJ0 (or @PRT IMSL*DOC.MMBSJ0) | Access: LIB NBS*IMSL

MMBSJ1 Bessel function of the first kind of order one. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C10a1 | Usage: D = MMBSJ1 (ARG,IER) | On-line doc: CALL GAMSDOC MMBSJ1 (or @PRT IMSL*DOC.MMBSJ1) | Access: LIB NBS*IMSL

MMBSJN Bessel function of the first kind of nonnegative integer order for real arguments. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C10a1 | Usage: CALL MMBSJN(ARG,N,B,IER) | On-line doc: CALL GAMSDOC MMBSJN (or @PRT IMSL*DOC.MMBSJN) | Access: LIB NBS*IMSL

- MMBSJR** Bessel function of the first kind of nonnegative real order for real positive arguments. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C10a3 | Usage: CALL MMBSJR(ARG,ORDER,N,RJ,WK,IER) | On-line doc: CALL GAMSDOC MMBSJR (or @PRT IMSL*DOC.MMBSJR) | Access: LIB NBS*IMSL
- MMBSK0** Modified Bessel function of the second kind of order zero. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C10b1 | Usage: D = MMBSK0 (IOPT,ARG,IER) | On-line doc: CALL GAMSDOC MMBSK0 (or @PRT IMSL*DOC.MMBSK0) | Access: LIB NBS*IMSL
- MMBSK1** Modified Bessel function of the second kind of order one. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C10b1 | Usage: D = MMBSK1 (IOPT,ARG,IER) | On-line doc: CALL GAMSDOC MMBSK1 (or @PRT IMSL*DOC.MMBSK1) | Access: LIB NBS*IMSL
- MMBSKR** Modified Bessel function of the second kind of nonnegative real fractional order for real positive arguments scaled by $\exp(\arg)$. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C10b3 | Usage: CALL MMBSKR(ARG,ORDER,N,BK,IER) | On-line doc: CALL GAMSDOC MMBSKR (or @PRT IMSL*DOC.MMBSKR) | Access: LIB NBS*IMSL
- MMBSYN** Bessel function of the second kind of nonnegative real fractional order for real positive arguments. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C10a3 | Usage: CALL MMBSYN (ARG,ORDER,N,YN,IER) | On-line doc: CALL GAMSDOC MMBSYN (or @PRT IMSL*DOC.MMBSYN) | Access: LIB NBS*IMSL
- MMBZIN** Modified Bessel function of the first kind of nonnegative integer order for complex arguments. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C10b2 | Usage: CALL MMBZIN(X,Y,N,BR,B1,IER) | On-line doc: CALL GAMSDOC MMBZIN (or @PRT IMSL*DOC.MMBZIN) | Access: LIB NBS*IMSL
- MMBZJN** Bessel function of the first kind of nonnegative integer order for complex arguments. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C10a2 | Usage: CALL MMBZJN(X,Y,N,BR,B1,IER) | On-line doc: CALL GAMSDOC MMBZJN (or @PRT IMSL*DOC.MMBZJN) | Access: LIB NBS*IMSL
- MMDAS** Dawson integral. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C8c | Usage: D = MMDAS (ARG,IER) | On-line doc: CALL GAMSDOC MMDAS (or @PRT IMSL*DOC.MMDAS) | Access: LIB NBS*IMSL
- MMDEI** Exponential integrals. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C5 | Usage: D = MMDEI (IOPT,ARG,IER) | On-line doc: CALL GAMSDOC MMDEI (or @PRT IMSL*DOC.MMDEI) | Access: LIB NBS*IMSL
- MMDELE** Complete elliptic integral of the second kind. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C14 | Usage: D = MMDELE (IOPT,ARG,IER) | On-line doc: CALL GAMSDOC MMDELE (or @PRT IMSL*DOC.MMDELE) | Access: LIB NBS*IMSL
- MMDELK** Complete elliptic integral of the first kind. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C14 | Usage: D = MMDELK (IOPT,ARG,IER) | On-line doc: CALL GAMSDOC MMDELK (or @PRT IMSL*DOC.MMDELK) | Access: LIB NBS*IMSL
- MMDEN** Exponential integrals of integer order for real argument x scaled by $\exp(x)$. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C5 | Usage: CALL MMDEN(X,N,F,IER) | On-line doc: CALL GAMSDOC MMDEN (or @PRT IMSL*DOC.MMDEN) | Access: LIB NBS*IMSL
- MMKEL0** Kelvin functions of the first kind, (ber,bei), and of the second kind, (ker,kei) of order zero. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C10c | Usage: CALL MMKEL0 (X,BER,BE1,XKER,XKE1,IER) | On-line doc: CALL GAMSDOC MMKEL0 (or @PRT IMSL*DOC.MMKEL0) | Access: LIB NBS*IMSL
- MMKEL1** Kelvin functions of the first kind, (ber,bei), and of the second kind, (ker,kei) of order one. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C10c | Usage: CALL MMKEL1 (X,BER1,BE11,XKER1,XKE11,IER) | On-line doc: CALL GAMSDOC MMKEL1 (or @PRT IMSL*DOC.MMKEL1) | Access: LIB NBS*IMSL
- MMKELD** Derivatives of the Kelvin functions (ber,bei,ker, and kei) of order zero. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C10c | Usage: CALL MMKELD (X,BERP,BEIP,XKERP,XKEIP,IER) | On-line doc: CALL GAMSDOC MMKELD (or @PRT IMSL*DOC.MMKELD) | Access: LIB NBS*IMSL
- MMLINC** Computes an elementary integral from which inverse circular functions, logarithms or inverse hyperbolic functions may be computed. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C4 | Usage: D = MMLINCC(X,Y,IER) | On-line doc: CALL GAMSDOC MMLINC (or @PRT IMSL*DOC.MMLINC) | Access: LIB NBS*IMSL
- MMLIND** Incomplete elliptic integral of the second kind. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C14 | Usage: D = MMLIND(X,Y,Z,IER) | On-line doc: CALL GAMSDOC MMLIND (or @PRT IMSL*DOC.MMLIND) | Access: LIB NBS*IMSL
- MMLINF** Incomplete elliptic integral of the first kind. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C14 | Usage: D = MMLINF(X,Y,Z,IER) | On-line doc: CALL GAMSDOC MMLINF (or @PRT IMSL*DOC.MMLINF) | Access: LIB NBS*IMSL
- MMLINJ** Incomplete elliptic integral of the third kind. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C14 | Usage: D = MMLINJ(X,Y,Z,P,IER) | On-line doc: CALL GAMSDOC MMLINJ (or @PRT IMSL*DOC.MMLINJ) | Access: LIB

NBS*IMSL

- MMPSI** Logarithmic derivative of the gamma function. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C7a | Usage: D = MMPSI (ARG,IER) | On-line doc: CALL GAMSDOC MMPSI (or @PRT IMSL*DOC.MMPSI) | Access: LIB NBS*IMSL
- MMWPL** Weierstrass p-function in the lemniscatic case for complex argument with unit period parallelogram. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C15 | Usage: CALL MMWPL(Z,PLEM,IER) | On-line doc: CALL GAMSDOC MMWPL (or @PRT IMSL*DOC.MMWPL) | Access: LIB NBS*IMSL
- MMWPL1** First derivative of the Weierstrass p-function in the lemniscatic case for complex argument with unit period parallelogram. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C15 | Usage: CALL MMWPL1(Z,PLEM,IER) | On-line doc: CALL GAMSDOC MMWPL1 (or @PRT IMSL*DOC.MMWPL1) | Access: LIB NBS*IMSL
- MMWPQ** Weierstrass p-function in the equianharmonic case for complex argument with unit period parallelogram. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C15 | Usage: CALL MMWPQ(Z,PEQ,IER) | On-line doc: CALL GAMSDOC MMWPQ (or @PRT IMSL*DOC.MMWPQ) | Access: LIB NBS*IMSL
- MMWPQ1** First derivative of the Weierstrass p-function in the equianharmonic case for complex argument with unit period parallelogram. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): C15 | Usage: CALL MMWPQ1(Z,PEQ1,IER) | On-line doc: CALL GAMSDOC MMWPQ1 (or @PRT IMSL*DOC.MMWPQ1) | Access: LIB NBS*IMSL
- MNPB** Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each interval. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DMNPB. | Class(es): E3 K6 | Usage: CALL MNPB (X,NX,N,K,T,NT) | On-line doc: CALL GAMSDOC MNPB (or @PRT PORT*DOC.MNPB) | Access: LIB NBS*PORT
- MOL1D** Solves systems of linear or nonlinear initial-boundary-value problems in one space dimension. Can solve hyperbolic equations with or without discontinuities, parabolic equations (including reaction-diffusion equations). Uses the method of lines based on equi-spaced finite differences. Graphical output available. | Portable single precision Fortran subprogram in PDELIB library. | Class(es): 12a1a 12a2 | Usage: CALL MOL1D(NPDE,NPTS,KEQN,KBC,METH,EPS,MORD,TINT,TLAST,MOUT,TOUT,UZ,XM,KMOL) | On-line doc: CALL GAMSDOC MOL1D (or @PRT PDELIB*DOC.MOL1D) | Tests: PDELIB*TEST-SOURCE.MOL1D | Access: LIB NBS*PDELIB
- MONOD** Test if a double precision vector is monotone increasing or decreasing. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is MONOR. | Class(es): R2 | Usage: L = MONOD (X, N, INC) | On-line doc: CALL GAMSDOC MONOD (or @PRT PORT*DOC.MONOD) | Access: LIB NBS*PORT
- MONOI** Test if an integer vector is monotone increasing or decreasing. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): R2 | Usage: L = MONOI (X, N, INC) | On-line doc: CALL GAMSDOC MONOI (or @PRT PORT*DOC.MONOI) | Access: LIB NBS*PORT
- MONOR** Test if a real vector is monotone increasing or decreasing. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is MONOD. | Class(es): R2 | Usage: L = MONOR (X, N, INC) | On-line doc: CALL GAMSDOC MONOR (or @PRT PORT*DOC.MONOR) | Access: LIB NBS*PORT
- MOVAVG** Computes a k-term symmetric moving average of a series. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L10b K5 | Usage: CALL MOVAVG (Y, N, K, H, YMA, YH, NYMA, YL, NYL) | On-line doc: CALL GAMSDOC MOVAVG (or @PRT STATLIB*DOC.MOVAVG) | Tests: STATLIB*TEST.DEMO3 | Access: LIB NBS*STATLIB
- MOVE** Copies M elements of the vector X (starting with position IX1) into the vector Y (starting with position IY1). | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): D1a5 | Usage: CALL MOVE(X,M,IX1,IY1,Y) | On-line doc: CALL GAMSDOC MOVE (or @PRT DATAPAC*DOC.MOVE) | Access: LIB NBS*DAPAC
- MOVEBC** Move a complex vector using backward DO loop. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): D1a5 | Usage: CALL MOVEBC (N,A,B) | On-line doc: CALL GAMSDOC MOVEBC (or @PRT PORT*DOC.MOVEBC) | Access: LIB NBS*PORT
- MOVEBD** Move a double precision vector using backward DO loop. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is MOVEBR. | Class(es): D1a5 | Usage: CALL MOVEBD (N,A,B) | On-line doc: CALL GAMSDOC MOVEBD (or @PRT PORT*DOC.MOVEBD) | Access: LIB NBS*PORT
- MOVEBI** Move an integer vector using backward DO loop. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): D1a5 | Usage: CALL MOVEBI (N,A,B) | On-line doc: CALL GAMSDOC MOVEBI (or @PRT PORT*DOC.MOVEBI) | Access: LIB NBS*PORT
- MOVEBL** Move a logical vector using backward DO loop. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): D1a5 | Usage: CALL MOVEBL (N,A,B) | On-line doc: CALL GAMSDOC MOVEBL (or @PRT PORT*DOC.MOVEBL) | Access: LIB NBS*PORT
- MOVEBR** Move a real vector using backward DO loop. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is MOVEBD. | Class(es): D1a5 | Usage: CALL MOVEBR (N,A,B) | On-line doc: CALL GAMSDOC MOVEBR (or @PRT PORT*DOC.MOVEBR) | Access: LIB NBS*PORT
- MOVEFC** Move a complex vector using forward DO loops. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): D1a5 | Usage: CALL MOVEFC (N,A,B) | On-line doc: CALL GAMSDOC MOVEFC (or @PRT PORT*DOC.MOVEFC) | Access: LIB

NBS*PORT

- MOVEFD** Move a double precision vector using forward DO loop. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): D1a5 | Usage: CALL MOVEFD (N,A,B) | On-line doc: CALL GAMSDOC MOVEFD (or @PRT PORT*DOC.MOVEFD) | Access: LIB NBS*PORT
- MOVEFI** Move an integer vector using forward DO loop. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): D1a5 | Usage: CALL MOVEFI (N,A,B) | On-line doc: CALL GAMSDOC MOVEFI (or @PRT PORT*DOC.MOVEFI) | Access: LIB NBS*PORT
- MOVEFL** Move a logical vector using forward DO loop. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): D1a5 | Usage: CALL MOVEFL (N,A,B) | On-line doc: CALL GAMSDOC MOVEFL (or @PRT PORT*DOC.MOVEFL) | Access: LIB NBS*PORT
- MOVEFR** Move a real vector using forward DO loop. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is MOVFRD. | Class(es): D1a5 | Usage: CALL MOVEFR (N,A,B) | On-line doc: CALL GAMSDOC MOVEFR (or @PRT PORT*DOC.MOVEFR) | Access: LIB NBS*PORT
- MPLT** Prints multiple scatter diagrams on the same axis. | Command in MINITAB Proprietary interactive system. Class(es): L3c3 Q1 | Usage: MPLT C vs C and C vs C [and ...] | On-line doc: HELP MPLT (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- MPLT** Displays a 50x100 character line printer plot of several dependent variables vs. a common independent variable. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c3 Q1 | Usage: CALL MPLT (YM, X, N, M, IYM) | On-line doc: CALL GAMSDOC MPLT (or @PRT STATLIB*DOC.MPLT) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB
- MPLTH** Displays a 50x50 character line printer plot of several dependent variables vs. a common independent variable. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c3 Q1 | Usage: CALL MPLTH (YM, X, N, M, IYM) | On-line doc: CALL GAMSDOC MPLTH (or @PRT STATLIB*DOC.MPLTH) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB
- MPLTHL** Displays a 50x50 character line printer plot of several dependent variables vs. a common independent variable, with user control of the plot limits. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c3 Q1 | Usage: CALL MPLTHL (YM, X, N, M, IYM, NOUT, YLB, YUB, XLB, XUB) | On-line doc: CALL GAMSDOC MPLTHL (or @PRT STATLIB*DOC.MPLTHL) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB
- MPLTL** Displays a 50x100 character line printer plot of several dependent variables vs. a common independent variable, with user control of the plot limits. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c3 Q1 | Usage: CALL MPLTL (YM, X, N, M, IYM, NOUT, YLB, YUB, XLB, XUB) | On-line doc: CALL GAMSDOC MPLTL (or @PRT STATLIB*DOC.MPLTL) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB
- MPOLISH** Uses median polish to fit an additive model to a two-way layout which may be unbalanced and may have empty cells. Options: fit columns first, number of iterations, save results. | Command in MINITAB Proprietary interactive system. Class(es): L9d | Usage: MPOLish the data in C, row levels in C, column levels in C [put residuals into C [fits into C]] [; COLUMNS first; ITERATIONS = K; EFFECTS, put common into K, row effects into C, columns effects into C; COMPARISON values, put into C.] | On-line doc: HELP MPOLISH (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- MSENO** Expected values of normal order statistics. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a2n | Usage: CALL MSENO (IFIRST, ILAST, N, EX, IER) | On-line doc: CALL GAMSDOC MSENO (or @PRT IMSL*DOC.MSENO) | Access: LIB NBS*IMSL
- MSMRAT** Ratio of the ordinate to the upper tail area of the standardized normal (Gaussian) distribution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1n | Usage: CALL MSMRAT (X,RM,IER) | On-line doc: CALL GAMSDOC MSMRAT (or @PRT IMSL*DOC.MSMRAT) | Access: LIB NBS*IMSL

N

- NAFRE** Friedman's test for randomised complete block designs. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a2a2 | Usage: CALL NAFRE (Y, NB, NT, ALPHA, STAT, RJ, D, IWK, WK, IER) | On-line doc: CALL GAMSDOC NAFRE (or @PRT IMSL*DOC.NAFRE) | Access: LIB NBS*IMSL
- NAK1** Kruskal-Wallis test for identical populations. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4b1b | Usage: CALL NAK1 (X,N1,M,EPS,IR,R,STAT,IER) | On-line doc: CALL GAMSDOC NAK1 (or @PRT IMSL*DOC.NAK1) | Access: LIB NBS*IMSL
- NAWNR** Wilsons ANOVA (2 or 3 way designs) without replicates. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a2a2 | Usage: CALL NAWNR (Y,N,IR,NF,NDF,STAT,IER) | On-line doc: CALL GAMSDOC NAWNR (or @PRT IMSL*DOC.NAWNR) | Access: LIB NBS*IMSL
- NAWRPE** Wilsons ANOVA (1, 2, or 3 way designs) with equal replication. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a1b L7a2a2 | Usage: CALL NAWRPE (Y,N,NREPS,IR,NF,NDF,STAT,IER) | On-line doc: CALL GAMSDOC NAWRPE (or @PRT IMSL*DOC.NAWRPE) | Access: LIB NBS*IMSL
- NAWRPU** Wilsons ANOVA (1, 2, or 3 way designs) with unequal replication. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a1b L7a4b | Usage: CALL NAWRPU (Y,N,NREPS,IR,NF,NDF,STAT,IER) | On-line doc: CALL GAMSDOC NAWRPU (or @PRT IMSL*DOC.NAWRPU) | Access: LIB NBS*IMSL
- NBCDF** Computes the cumulative distribution function value at X for the negative binomial distribution with parameters P and N. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1n | Usage: CALL NBCDF(X,P,N,CDF) | On-line doc: CALL GAMSDOC NBCDF (or @PRT DATAPAC*DOC.NBCDF) | Access: LIB NBS*DATAPAC
- NBCYC** Noethers test for cyclical trend. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1d | Usage: CALL NBCYC (X,N,EPS,NSTAT,P,IER) | On-line doc: CALL GAMSDOC NBCYC (or @PRT IMSL*DOC.NBCYC) | Access: LIB NBS*IMSL
- NBPPF** Computes the percent point function value at P for the negative binomial distribution with parameters PPAR and N. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2n | Usage: CALL NBPPF(P,PPAR,N,PPF) | On-line doc: CALL GAMSDOC NBPPF (or @PRT DATAPAC*DOC.NBPPF) | Access: LIB NBS*DATAPAC
- NBQT** Cochran q test. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4b1b | Usage: CALL NBQT (X,N,M,IA,Q,PQ,IER) | On-line doc: CALL GAMSDOC NBQT (or @PRT IMSL*DOC.NBQT) | Access: LIB NBS*IMSL
- NBRAN** Generates a random sample of size N from the negative binomial distribution with parameters P and NPAR. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a14 | Usage: CALL NBRAN(N,P,NPAR,ISTART,X) | On-line doc: CALL GAMSDOC NBRAN (or @PRT DATAPAC*DOC.NBRAN) | Access: LIB NBS*DATAPAC
- NBSDL** Cox and Stuart sign test for trends in dispersion and location. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1d | Usage: CALL NBSDL (IOPT,X,N,K,IDS,EPS,NSTAT,PSTAT,IER) | On-line doc: CALL GAMSDOC NBSDL (or @PRT IMSL*DOC.NBSDL) | Access: LIB NBS*IMSL
- NBSIGN** Sign test (for percentiles). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1b | Usage: CALL NBSIGN (X,N,Q,P,NSIGN,PROB,IER) | On-line doc: CALL GAMSDOC NBSIGN (or @PRT IMSL*DOC.NBSIGN) | Access: LIB NBS*IMSL
- NDEST** Evaluate probability density function at specified points. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1g | Usage: CALL NDEST (Y,N,F,M,IOPT,B,WK,EST,IER) | On-line doc: CALL GAMSDOC NDEST (or @PRT IMSL*DOC.NDEST) | Access: LIB NBS*IMSL
- NDKER** Nonparametric probability density function (one dimensional) estimation by the kernel method. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1e | Usage: CALL NDKER (X,N,H,DEL,XKER,B,NPT,F) | On-line doc: CALL GAMSDOC NDKER (or @PRT IMSL*DOC.NDKER) | Access: LIB NBS*IMSL
- NDMPLE** Nonparametric probability density function (one dimensional) estimation by the penalized likelihood method. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1e | Usage: CALL NDMPLE (X,N,NMAX,IND,ALPHA,EPS,ITMAX,F,B,XJAC,IXJAC,ILOHI, DELC,IER) | On-line doc: CALL GAMSDOC NDMPLE (or @PRT IMSL*DOC.NDMPLE) | Access: LIB NBS*IMSL
- NDRV** Solves sparse systems of linear algebraic equations by Gaussian elimination without pivoting (uncompressed storage mode). | Portable single precision Fortran subprogram in YSMP sublibrary of CMLIB library. | Class(es): D2a4 | Usage: CALL NDRV(N,R,C,IC,IA,JA,A,B,Z,NSP,RSP,ESP,PATH,FLAG) | On-line doc: CALL GAMSDOC NDRV (or @PRT CMLIB*DOC.NDRV/YSMP) | Tests: CMLIB*TEST-SOURCE.\$Q2/YSMP | Access: LIB NBS*CMLIB
- NERROR** Provides the current error number for PORT library programs. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): R3c | Usage: I = NERROR (NERR) | On-line doc: CALL GAMSDOC NERROR (or @PRT PORT*DOC.NERROR) | Access: LIB NBS*PORT
- NFIT** Performs nonlinear least squares regression analysis using the Levenberg-Marquardt algorithm. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L8g1a | Usage: CALL NFIT (Y, XM, MODEL, N, NP, NX, SCRAT, NS, IXM, RES, COEF) |

- On-line doc:** CALL GAMSDOC NFIT (or @PRT STATLIB*DOC.NFIT) | **Tests:** STATLIB*TEST.DEMO2 | **Access:** LIB NBS*STATLIB
- NFITS** Performs nonlinear least squares regression analysis using the Levenberg-Marquardt algorithm, with results returned to the user. | Portable single precision Fortran subprogram in STATLIB library. | **Class(es):** L8g1a | **Usage:** CALL NFITS (Y, XM, MODEL, N, NP, NX, SCRAT, NS, IXM, RES, COEF, STAT, PV, SDPV, SDRES, VCV, IVCV, FLAMBD, FNU, MIT, STEP, STOPSS, STOPCR, NPRT, IEIG) | **On-line doc:** CALL GAMSDOC NFITS (or @PRT STATLIB*DOC.NFITS) | **Tests:** STATLIB*TEST.DEMO2 | **Access:** LIB NBS*STATLIB
- NFITW** Performs weighted nonlinear least squares regression analysis using the Levenberg-Marquardt algorithm. | Portable single precision Fortran subprogram in STATLIB library. | **Class(es):** L8g2a | **Usage:** CALL NFITW (Y, XM, WT, MODEL, N, NP, NX, SCRAT, NS, IXM, RES, COEF) | **On-line doc:** CALL GAMSDOC NFITW (or @PRT STATLIB*DOC.NFITW) | **Tests:** STATLIB*TEST.DEMO2 | **Access:** LIB NBS*STATLIB
- NFITWS** Performs weighted nonlinear least squares regression analysis using the Levenberg-Marquardt algorithm, with results returned to the user. | Portable single precision Fortran subprogram in STATLIB library. | **Class(es):** L8g2a | **Usage:** CALL NFITWS (Y, XM, WT, MODEL, N, NP, NX, SCRAT, NS, IXM, RES, COEF, STAT, PV, SDPV, SDRES, VCV, IVCV, FLAMBD, FNU, MIT, STEP, STOPSS, STOPCR, NPRT, IEIG) | **On-line doc:** CALL GAMSDOC NFITWS (or @PRT STATLIB*DOC.NFITWS) | **Tests:** STATLIB*TEST.DEMO2 | **Access:** LIB NBS*STATLIB
- NHEXT** Fishers exact method for 2-by-2 tables. | Proprietary single precision Fortran subprogram in IMSL library. | **Class(es):** L9a | **Usage:** CALL NHEXT (ITAB,IT,P,IER) | **On-line doc:** CALL GAMSDOC NHEXT (or @PRT IMSL*DOC.NHEXT) | **Access:** LIB NBS*IMSL | See also: BDCOU1 BDCOU2
- NHINC** Includance test. | Proprietary single precision Fortran subprogram in IMSL library. | **Class(es):** L4b1b | **Usage:** CALL NHINC (X,N,Y,M,I1,I2,STAT,IER) | **On-line doc:** CALL GAMSDOC NHINC (or @PRT IMSL*DOC.NHINC) | **Access:** LIB NBS*IMSL
- NKS1** Kolmogorov-Smirnov one-sample test. | Proprietary single precision Fortran subprogram in IMSL library. | **Class(es):** L4a1c | **Usage:** CALL NKS1 (PDF,X,N,PDIF,IER) | **On-line doc:** CALL GAMSDOC NKS1 (or @PRT IMSL*DOC.NKS1) | **Access:** LIB NBS*IMSL
- NKS2** Kolmogorov-Smirnov two-sample test. | Proprietary single precision Fortran subprogram in IMSL library. | **Class(es):** L4b1b | **Usage:** CALL NKS2 (F,N,G,M,PDIF,IER) | **On-line doc:** CALL GAMSDOC NKS2 (or @PRT IMSL*DOC.NKS2) | **Access:** LIB NBS*IMSL
- NL2S1** Minimizes a nonlinear sum of squares using both residual and gradient values supplied by the user. | Portable single precision Fortran subprogram in NL2SN sublibrary of CMLIB library. Double precision version is DNL2S1. | **Class(es):** L8g1b L8g2b K1b1a2 | **Usage:** CALL NL2S1(N,P,X,CALCR,CALCJ,IV,LIV,LV,V,UIPARM,URPARM,UFPARM) | **On-line doc:** CALL GAMSDOC NL2S1 (or @PRT CMLIB*DOC.NL2S1/NL2SN) | **Tests:** CMLIB*TEST-SOURCE.\$F1/NL2SN, CMLIB*TEST-SOURCE.\$Q1/NL2SN | **Access:** LIB NBS*CMLIB
- NL2SN** Minimizes a nonlinear sum of squares using residual values only. | Portable single precision Fortran subprogram in NL2SN sublibrary of CMLIB library. Double precision version is DNL2SN. | **Class(es):** L8g1a L8g2a K1b1a1 | **Usage:** CALL NL2SN(N,P,X,CALCR,IV,LIV,LV,V,UIPARM,URPARM,UFPARM) | **On-line doc:** CALL GAMSDOC NL2SN (or @PRT CMLIB*DOC.NL2SN/NL2SN) | **Tests:** CMLIB*TEST-SOURCE.\$F1/NL2SN, CMLIB*TEST-SOURCE.\$Q1/NL2SN | **Access:** LIB NBS*CMLIB
- NMCC** Calculate and test the significance of the Kendall coefficient of concordance. | Proprietary single precision Fortran subprogram in IMSL library. | **Class(es):** L4b1b | **Usage:** CALL NMCC (X,M,N,IX,EPS,IR,R,STAT,IER) | **On-line doc:** CALL GAMSDOC NMCC (or @PRT IMSL*DOC.NMCC) | **Access:** LIB NBS*IMSL
- NMKN** Kendall's test for correlation (rank correlation coefficient). | Proprietary single precision Fortran subprogram in IMSL library. | **Class(es):** L4b1b | **Usage:** CALL NMKN (X, Y, N, STAT, XSTAT, IWK, WK, IER) | **On-line doc:** CALL GAMSDOC NMKN (or @PRT IMSL*DOC.NMKN) | **Access:** LIB NBS*IMSL
- NMKSF** Frequency distribution of K and the probability of equalling or exceeding K, where K, the total score from the Kendall rank correlation coefficient calculations, and N, the sample size, are given. | Proprietary single precision Fortran subprogram in IMSL library. | **Class(es):** L4b1b | **Usage:** CALL NMKSF (K,N,Z,ZW,P) | **On-line doc:** CALL GAMSDOC NMKSF (or @PRT IMSL*DOC.NMKSF) | **Access:** LIB NBS*IMSL | See also: NMKN
- NMKTS** K-sample trends test against ordered alternatives. | Proprietary single precision Fortran subprogram in IMSL library. | **Class(es):** L4b1b | **Usage:** CALL NMKTS (X,XM,K,DSEED,XSTAT,IER) | **On-line doc:** CALL GAMSDOC NMKTS (or @PRT IMSL*DOC.NMKTS) | **Access:** LIB NBS*IMSL
- NMRANK** Numerical ranking. | Proprietary single precision Fortran subprogram in IMSL library. | **Class(es):** N6a1b1 | **Usage:** CALL NMRANK (X,N,EPS,IR,R,RANK,S,T) | **On-line doc:** CALL GAMSDOC NMRANK (or @PRT IMSL*DOC.NMRANK) | **Access:** LIB NBS*IMSL
- NMTIE** Tie statistics, given a sample of observations. | Proprietary single precision Fortran subprogram in IMSL library. | **Class(es):** L1a1 | **Usage:** CALL NMTIE (X,M,EPS,TIES) | **On-line doc:** CALL GAMSDOC NMTIE (or @PRT IMSL*DOC.NMTIE) | **Access:** LIB NBS*IMSL
- NORCDF** Computes the cumulative distribution function value for the normal (Gaussian) distribution with mean = 0 and standard deviation = 1. | Portable single precision Fortran subprogram in DATAPAC library. | **Class(es):** L5a1n | **Usage:** CALL NORCDF(X,CDF) | **On-line doc:** CALL GAMSDOC NORCDF (or @PRT DATAPAC*DOC.NORCDF) | **Access:** LIB NBS*DAPAC
- NOROUT** Performs a normal outlier analysis on the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | **Class(es):** L4a1a14 | **Usage:** CALL NOROUT(X,N) | **On-line doc:** CALL GAMSDOC NOROUT (or @PRT

DATAPAC*DOC.NOROUT) | Access: LIB NBS*DATAPAC

NORPDF Computes the probability density function value for the normal (Gaussian) distribution with mean = 0 and standard deviation = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1n | Usage: CALL NORPDF(X,PDF) | On-line doc: CALL GAMSDOC NORPDF (or @PRT DATAPAC*DOC.NORPDF) | Access: LIB NBS*DATAPAC

NORPLT Generates a normal (Gaussian) probability plot with mean = 0 and standard deviation = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c4n | Usage: CALL NORPLT(X,N) | On-line doc: CALL GAMSDOC NORPLT (or @PRT DATAPAC*DOC.NORPLT) | Access: LIB NBS*DATAPAC

NORPPF Computes the percent point function value for the normal (Gaussian) distribution with mean = 0 and standard deviation = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2n | Usage: CALL NORPPF(P,PPF) | On-line doc: CALL GAMSDOC NORPPF (or @PRT DATAPAC*DOC.NORPPF) | Access: LIB NBS*DATAPAC

NORRAN Generates a random sample of size N from the normal (Gaussian) distribution with mean = 0 and standard deviation = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a14 | Usage: CALL NORRAN(N,ISTART,X) | On-line doc: CALL GAMSDOC NORRAN (or @PRT DATAPAC*DOC.NORRAN) | Access: LIB NBS*DATAPAC

NORSF Computes the sparsity function value for the normal (Gaussian) distribution with mean = 0 and standard deviation = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2n | Usage: CALL NORSF(P,SF) | On-line doc: CALL GAMSDOC NORSF (or @PRT DATAPAC*DOC.NORSF) | Access: LIB NBS*DATAPAC

NRAND Generates a vector of normally distributed pseudo-random numbers. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L6a14 | Usage: CALL NRAND (Y, N, ISEED) | On-line doc: CALL GAMSDOC NRAND (or @PRT STATLIB*DOC.NRAND) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

NRANDOM Generates K pseudo-random numbers from the normal distribution with specified mean and standard deviation. | Command in MINITAB Proprietary interactive system. Class(es): L6a14 | Usage: NRANDOM K observations with mu = K, sigma = K, put into C | On-line doc: HELP NRANDOM (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

NRBHA Bhapkar v test. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4b1b | Usage: CALL NRBHA (X,NPS,IC,IR,W,V,Q,IER) | On-line doc: CALL GAMSDOC NRBHA (or @PRT IMSL*DOC.NRBHA) | Access: LIB NBS*IMSL

NRWMD Wilcoxon signed rank test. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4b1b | Usage: CALL NRWMD (IOPT,X,Y,N,EPS,DSEED,IR,STAT,IER) | On-line doc: CALL GAMSDOC NRWMD (or @PRT IMSL*DOC.NRWMD) | Access: LIB NBS*IMSL

NRWRST Wilcoxon's rank-sum test. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4b1b | Usage: CALL NRWRST (X,M,N,EPS,IR,STAT,IER) | On-line doc: CALL GAMSDOC NRWRST (or @PRT IMSL*DOC.NRWRST) | Access: LIB NBS*IMSL

NSCORES Calculates normal scores. | Command in MINITAB Proprietary interactive system. Class(es): L2a | Usage: NSCORES of C, put into C | On-line doc: HELP NSCORES (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

NUMXER Get most current message number. | Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: I=NUMXER(NERR) | On-line doc: CALL GAMSDOC NUMXER (or @PRT CMLIB*DOC.SUMMARY/XERROR) | Access: LIB NBS*CMLIB

O

- OCDIS** Pairwise Euclidean distances between the columns of a matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L14a1a | Usage: CALL OCDIS (X,IX,N,M,WK,XDIS) | On-line doc: CALL GAMSDOC OCDIS (or @PRT IMSL*DOC.OCDIS) | Access: LIB NBS*IMSL
- OCLINK** Perform a single-linkage or complete-linkage hierarchical cluster analysis given a similarity matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L14a1a | Usage: CALL OCLINK (ND,IOP,T,XSIM,CLEVEL,ICLSON,ICRSON,IPTR,IER) | On-line doc: CALL GAMSDOC OCLINK (or @PRT IMSL*DOC.OCLINK) | Access: LIB NBS*IMSL
- ODEQ** Finds the integral of a set of functions over the same interval by using the differential equation solver ODES1. For smooth functions. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DODEQ. | Class(es): H2a1a1 | Usage: CALL ODEQ (N, F, A, B, EPS, ANS) | On-line doc: CALL GAMSDOC ODEQ (or @PRT PORT*DOC.ODEQ) | Access: LIB NBS*PORT
- ODES** Solves an initial value problem for a system of ordinary differential equations. Easy to use. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DODES. | Class(es): 11a1c | Usage: CALL ODES (F,X,NX,TSTART,TSTOP,DT,ERRPAR,HANDLE) | On-line doc: CALL GAMSDOC ODES (or @PRT PORT*DOC.ODES) | Access: LIB NBS*PORT
- ODES1** Solves an initial value problem for a system of ordinary differential equations. Allows great flexibility and user control. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DODES1. | Class(es): 11a1c | Usage: CALL ODES1 (F,X,NX,TSTART,TSTOP,DT,ERROR,ERRPAR,HANDLE,GLBMAX, ERPUTS) | On-line doc: CALL GAMSDOC ODES1 (or @PRT PORT*DOC.ODES1) | Access: LIB NBS*PORT | See also: ODESH ODESE
- ODESE** Standard error subprogram for the routine ODES1. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DODESE. | Class(es): 11c | Usage: L = ODESE (X,NX,T,DT,ERRPAR,ERPUTS,E) | On-line doc: CALL GAMSDOC ODESE (or @PRT PORT*DOC.ODESE) | Access: LIB NBS*PORT | See also: ODES1
- ODESH** Default HANDLE routine for ODES. Used to access the results at the end of each integration time step. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DODESH. | Class(es): 11c | Usage: CALL ODESH (T0,X0,T1,X1,NX,DT,TSTOP,E) | On-line doc: CALL GAMSDOC ODESH (or @PRT PORT*DOC.ODESH) | Access: LIB NBS*PORT | See also: ODES
- ODFISH** Linear discriminant analysis method of Fisher for reducing the number of variables. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L12 | Usage: CALL ODFISH (X,IX,NG,NV,ND,XM,IXM,NNV,E,C,IC,SW,SB,ISB,EX,CX, ICX,IS,IER) | On-line doc: CALL GAMSDOC ODFISH (or @PRT IMSL*DOC.ODFISH) | Access: LIB NBS*IMSL
- ODNORM** Multivariate normal linear discriminant analysis among several known groups. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L12 | Usage: CALL ODNORM (X,IX,NB,NV,ND,P,XM,IXM,S,W,IW,D,ID,IER) | On-line doc: CALL GAMSDOC ODNORM (or @PRT IMSL*DOC.ODNORM) | Access: LIB NBS*IMSL
- ODRV** Computes the minimum degree ordering of equations and unknowns for a system of linear algebraic equations in sparse storage mode. | Portable single precision Fortran subprogram in YSMP sublibrary of CMLIB library. | Class(es): D2b4 | Usage: CALL ODRV(N,IA,JA,A,P,IP,NSP,ISP,PATH,FLAG) | On-line doc: CALL GAMSDOC ODRV (or @PRT CMLIB*DOC.ODRV/YSMP) | Tests: CMLIB*TEST-SOURCE.\$Q1/YSMP, CMLIB*TEST-SOURCE.\$Q2/YSMP | Access: LIB NBS*CMLIB
- OFCOEF** Compute a matrix of factor score coefficients for input to IMSL routine OFSCOR. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L13 | Usage: CALL OFCOEF (B,IB,NV,NF,IND,S,T,IT,C,IC,IS,WK,IER) | On-line doc: CALL GAMSDOC OFCOEF (or @PRT IMSL*DOC.OFCOEF) | Access: LIB NBS*IMSL
- OFCOMM** Compute an unrotated factor loading matrix according to a common factor model by unweighted or generalized least sqrs., or by max. likelihood procedures. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L13 | Usage: CALL OFCOMM (R,NV,NF,IND,NT,IV,MAXIT,MAXTRY,EPS,EPSE,ALPHA,V,A, IA,RI,Y,S,G,IS,WK) | On-line doc: CALL GAMSDOC OFCOMM (or @PRT IMSL*DOC.OFCOMM) | Access: LIB NBS*IMSL
- OFHARR** Transformation of unrotated factor loading matrix to oblique axes by Harris-Kaiser method. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L13 | Usage: CALL OFHARR (A,IA,NV,NF,NORM,II,MAXIT,W,EPS,DELTA,C,G,B,IB,T, IT,F,S,WK,IER) | On-line doc: CALL GAMSDOC OFHARR (or @PRT IMSL*DOC.OFHARR) | Access: LIB NBS*IMSL
- OFIMA3** Least squares solution to the matrix equation $AT = B$. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a8 D9 | Usage: CALL OFIMA3 (A,IA,B,IB,NV,NS,NF,T,IT,WK,IER) | On-line doc: CALL GAMSDOC OFIMA3 (or @PRT IMSL*DOC.OFIMA3) | Access: LIB NBS*IMSL
- OFIMAG** Compute an unrotated factor loading matrix according to an image model. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L13 | Usage: CALL OFIMAG (R,NV,NF,A,IA,CI,IC,Y,S,G,IS,WK,IER) | On-line doc: CALL GAMSDOC OFIMAG (or @PRT IMSL*DOC.OFIMAG) | Access: LIB NBS*IMSL
- OFPRI** Compute an unrotated factor loading matrix according to a principal component model. | Proprietary single precision Fortran

- subprogram in IMSL library. | Class(es): L13a | Usage: CALL OFFPRI (R,NV,NF,CRIT,A,IA,E,Y,S,G,IS,WK,IER) | On-line doc: CALL GAMSDOC OFFPRI (or @PRT IMSL*DOC.OFFPRI) | Access: LIB NBS*IMSL
- OFFROT** Oblique transformation of the factor loading matrix using a target matrix, including pivot and power vector options. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L13 | Usage: CALL OFFROT (A,IA,NV,NF,IND,NORM,II,MAXIT,W,EPS,DELTA,F,X,IX,B, IB,T,IT,S,WK,IER) | On-line doc: CALL GAMSDOC OFFROT (or @PRT IMSL*DOC.OFFROT) | Access: LIB NBS*IMSL
- OFRESI** Communalities and normalized factor residual correlation matrix calculation. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L13 | Usage: CALL OFRESI (R,NV,NF,A,IA,Y,S,WK) | On-line doc: CALL GAMSDOC OFRESI (or @PRT IMSL*DOC.OFRESI) | Access: LIB NBS*IMSL
- OFROTA** Orthogonal rotation of a factor loading matrix using a generalized orthomax criterion, including quartimax, varimax, and equamax. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L13 | Usage: CALL OFROTA (A,IA,NV,NF,NORM,II,MAXIT,W,EPS,DELTA,B,IB,T,IT,F, WK,IER) | On-line doc: CALL GAMSDOC OFROTA (or @PRT IMSL*DOC.OFROTA) | Access: LIB NBS*IMSL
- OFSCHN** Orthogonal transformation of the factor loading matrix using a target matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L13 | Usage: CALL OFSCHN (A,IA,NV,NF,II,X,IX,B,IB,T,IT,S,F,IS,WK,IER) | On-line doc: CALL GAMSDOC OFSCHN (or @PRT IMSL*DOC.OFSCHN) | Access: LIB NBS*IMSL
- OFSCOR** Compute a set of factor scores given the factor score coefficient matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L13 | Usage: CALL OFSCOR (C,IC,NV,NF,NT,Z,IZ,ZBAR,STD,FMEAN,SS,WK,IER) | On-line doc: CALL GAMSDOC OFSCOR (or @PRT IMSL*DOC.OFSCOR) | Access: LIB NBS*IMSL | See also: OFCOEF
- OIND** Wilks test for the independence of k sets of multi-normal variates. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4c4a14 | Usage: CALL OIND (S,N,IP,K,STAT,WKAREA,IER) | On-line doc: CALL GAMSDOC OIND (or @PRT IMSL*DOC.OIND) | Access: LIB NBS*IMSL
- ONEWAY** Performs one-way analysis of variance, prints standard results, and optionally saves results. | Command in MINITAB Proprietary interactive system. Class(es): L7a1a | Usage: ONEWay analysis of variance for data in C, subscripts in C [put residuals into C [fits into C]] | On-line doc: HELP ONEWAY (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- ONEWY** Performs one-way analysis of variance for two or more random samples. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L7a1a | Usage: CALL ONEWY (Y, TAG, N, SCRAT, NS) | On-line doc: CALL GAMSDOC ONEWY (or @PRT STATLIB*DOC.ONEWY) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB
- ONEWYS** Performs one-way analysis of variance for two or more random samples, with computed results returned to the user. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L7a1a | Usage: CALL ONEWYS (Y, TAG, N, SCRAT, NS, NPRT, TAGVAL, GRPNOB, GRPAVG, GRPSD) | On-line doc: CALL GAMSDOC ONEWYS (or @PRT STATLIB*DOC.ONEWYS) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB
- OPRINC** Principal components of a multivariate sample of observations. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L13a | Usage: CALL OPRINC (S,M,IA,EVAL,EVEC,COMP,VAR,CL,CU,IER) | On-line doc: CALL GAMSDOC OPRINC (or @PRT IMSL*DOC.OPRINC) | Access: LIB NBS*IMSL
- ORDER** Sorts in ascending order the values in each of one or more vectors. | Command in MINITAB Proprietary interactive system. Class(es): N6a2b1 | Usage: ORDER C [and C,...,C] put into C [and into C,...,C] | On-line doc: HELP ORDER (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- ORTBAK** Forms eigenvectors of general real matrix from eigenvectors of upper Hessenberg matrix output from ORTHES. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL ORTBAK(NM,LOW,IGH,A,ORT,M,Z) | On-line doc: CALL GAMSDOC ORTBAK (or @PRT CMLIB*DOC.ORTBAK/EISPACK) | Access: LIB NBS*CMLIB | See also: ORTHES
- ORTHES** Reduces real general matrix to upper Hessenberg form using orthogonal similarity transformations. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1b2 | Usage: CALL ORTHES(NM,N,LOW,IH1,A,ORT) | On-line doc: CALL GAMSDOC ORTHES (or @PRT CMLIB*DOC.ORTHES/EISPACK) | Access: LIB NBS*CMLIB
- ORTHP** Evaluates a polynomial expressed as a sum of general orthogonal polynomials. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DORTHP. | Class(es): C3 | Usage: X = ORTHP (N, ALPHA, X, A, B, C) | On-line doc: CALL GAMSDOC ORTHP (or @PRT PORT*DOC.ORTHP) | Access: LIB NBS*PORT
- ORTRAN** Accumulates orthogonal similarity transformations in reduction of real general matrix by ORTHES. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL ORTRAN(NM,N,LOW,IH1,A,ORT,Z) | On-line doc: CALL GAMSDOC ORTRAN (or @PRT CMLIB*DOC.ORTRAN/EISPACK) | Access: LIB NBS*CMLIB | See also: ORTHES
- OTMLNR** Maximum likelihood estimation from grouped and/or censored normal data. | Proprietary single precision Fortran subprogram in

IMSL library. | Class(es): L4a3a14 | Usage: CALL OTMLNR (X,IXI,N,IP,RM,SIGMA,E1,E2,MAXITS,IDS,COV,NOBS,K, IER) | On-line
doc: CALL GAMSDOC OTMLNR (or @PRT IMSL*DOC.OTMLNR) | Access: LIB NBS*IMSL

P

- P01AAE** Return value of error indicator, or terminate with an error message. Used exclusively by NAG library programs. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is P01AAF. | Class(es): R3c | Usage: 1 = P01AAE (IFAIL, IERROR, SRNAME) | On-line doc: CALL GAMSDOC P01AAE (or @PRT NAG*DOC.P01AAE) | Access: LIB NBS*NAG
- P01AAF** Return value of error indicator, or terminate with an error message. Used exclusively by NAG library programs. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is P01AAE. | Class(es): R3c | Usage: 1 = P01AAF (IFAIL, IERROR, SRNAME) | On-line doc: CALL GAMSDOC P01AAF (or @PRT NAG*DOC.P01AAF) | Access: LIB NBS*NAG
- P1D** Prints univariate statistics (mean, standard deviation, std. error of mean, coefficient of variation, extreme values, extreme z-scores, range) for each variable. Options: statistics for each level of each grouping variable, sorting, printing all cases OR only cases with values missing or values outside specified limits. | Proprietary stand-alone program using BMDP command language. | Class(es): L1a1 L1a2 | Usage: @NBS*PLIB\$.BMDP BMDP1D | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP1D | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P1L** Estimates survival (time-to-response) distribution of patients who have been observed over varying periods of time by product-limit (Kaplan-Meier) or actuarial life table (Cutler-Ederer) method. Options: three forms of input, Mantel-Cox and Breslow test of equality of survival curves, five plots. | Proprietary stand-alone program using BMDP command language. | Class(es): L15 | Usage: @NBS*PLIB\$.BMDP BMDP1L | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP1L | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P1M** Stepwise cluster analysis of variables using one of four measures of similarity, three criteria for combining clusters, with printing of a summary table of clusters, shaded distance measure display, and a tree showing cluster formation. Options: form of input, additional printing and display. | Proprietary stand-alone program using BMDP command language. | Class(es): L14a1a | Usage: @NBS*PLIB\$.BMDP BMDP1M | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP1M | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P1R** Performs multiple linear regression and prints standard results. Options: weights, form of input, regression on subsets or groups and test of equality of regression lines, intercept term present or absent, more printing, five plots, save predicted values and residuals. | Proprietary stand-alone program using BMDP command language. | Class(es): L8a4 L8a7 | Usage: @NBS*PLIB\$.BMDP BMDP1R | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP1R | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P1S** At each pass through the data, computes univariate statistics (choose means, standard deviations, geometric means, harmonic means, extreme values), and transforms or edits the data using statistics computed in the previous pass. Options: printing, save results. | Proprietary stand-alone program using BMDP command language. | Class(es): L2a | Usage: @NBS*PLIB\$.BMDP BMDP1S | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP1S | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P1T** Interactive or batch spectral analysis of one or two time series, with estimates of spectral density and coherence between variables. Options: print, plot (variable vs. time, lagged plots, complex demodulation, periodogram), handle missing values, remove seasonal means and linear trend, filtering, save results. | Proprietary stand-alone program using BMDP command language. | Class(es): L10f | Usage: @NBS*PLIB\$.BMDP BMDP1T | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP1T | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P1V** Performs one-way ANOVA or ANCOVA with standard results. For ANCOVA, tests 1) equality of slopes, 2) zero slope, and 3) equality of adjusted cell means; plots the covariate for each group. Tests equality of pairs of means (or adjusted means). Options: linear contrasts, within-group correlations and statistics. | Proprietary stand-alone program using BMDP command language. | Class(es): L7a1a L7a3 | Usage: @NBS*PLIB\$.BMDP BMDP1V | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP1V | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P2D** For each variable, prints frequency and percent for each distinct value; mean, median, mode, standard deviation, std. errors of mean and median, skewness, kurtosis, half interquartile range; histogram, and stem-and-leaf plot. Options: initially round or truncate, three robust location estimates. | Proprietary stand-alone program using BMDP command language. | Class(es): L1a1 L1a2 | Usage: @NBS*PLIB\$.BMDP BMDP2D | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP2D | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P2L** Analyzes survival data with covariates using Cox proportional hazard regression model. Options: two forms of input, stepwise selection of covariates, time-dependent covariates, stratification, significance tests, three plots, print survival functions and residuals. | Proprietary stand-alone program using BMDP command language. | Class(es): L15 | Usage: @NBS*PLIB\$.BMDP BMDP2L | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP2L | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P2M** Stepwise cluster analysis of cases (observations) using one of four distance measures (including Euclidean and one for data that are frequency counts) and three linkage algorithms (single, centroid, k nearest neighbors), with a summary table of clusters and a cluster tree. Options: weights, standardized data. | Proprietary stand-alone program using BMDP command language. | Class(es): L14a1a | Usage: @NBS*PLIB\$.BMDP BMDP2M | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP2M | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P2R** Multiple linear regression, with standard results. Options: weights, forward or backward stepping, interactive stepping, stepping sets of variables (e.g. design variables), forcing variables into the model, eleven diagnostics (including Cook and AP statistics) available for printing, plotting, and saving. | Proprietary stand-alone program using BMDP command language. | Class(es): L8a4 L8a9 L8a5 | Usage: @NBS*PLIB\$.BMDP BMDP2R | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP2R | Access: @NBS*PLIB\$.BMDP BMD<program name>

- P2T** Interactive or batch Box-Jenkins time series analysis for univariate time domain models (including ARIMA, regression, intervention, and transfer function models) – model identification, parameter estimation, testing, forecasting. Options: print, plot, differencing and filtering, save results. | Proprietary stand-alone program using BMDP command language. | Class(es): L10c | Usage: @NBS*PLIB\$.BMDP BMDP2T | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP2T | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P2V** ANOVA and ANCOVA for unbalanced fixed-effect models (including full and fractional factorial designs, Latin squares), and repeated measure models, or a combination of models, with Greenhouse-Geisser and Huynh-Feldt deg. of freedom adjustment. Options: orthogonal decomposition of within-effects, save results. | Proprietary stand-alone program using BMDP command language. | Class(es): L7a | Usage: @NBS*PLIB\$.BMDP BMDP2V | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP2V | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P3D** One-sample t-test to test if one group mean is zero (e.g., matched pairs); two-sample t test with and without equal variances assumption, Levene's test for equal variances, histograms. Options: trimmed t test, Hotelling's T-squared and Mahalanobis' D-squared, within-group correlations, data listing. | Proprietary stand-alone program using BMDP command language. | Class(es): L4a L4b | Usage: @NBS*PLIB\$.BMDP BMDP3D | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP3D | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P3M** Forms blocks (submatrices of the data matrix) where a subset of the cases (for a subset of the variables) cluster together, with printing of the blocks and tree diagrams for cases and for variables – appropriate for categorical data with few levels. | Proprietary stand-alone program using BMDP command language. | Class(es): L14a2 | Usage: @NBS*PLIB\$.BMDP BMDP3M | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP3M | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P3R** Performs nonlinear least squares regression with standard results. Six functions are built in; others can be specified. Options: weights, evaluates functions of parameters (with standard errors), upper and lower limits on parameters, ridging, exact linear constraints, maximum likelihood estimates, and five plots. | Proprietary stand-alone program using BMDP command language. | Class(es): L8g1b L8g2b L8b | Usage: @NBS*PLIB\$.BMDP BMDP3R | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP3R | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P3S** Computes and prints results from one or more of the following: sign test, Wilcoxon signed-rank test, Mann-Whitney rank-sum test, Kruskal-Wallis one-way ANOVA, Friedman two-way ANOVA, Kendall's coefficient of concordance, Kendall and Spearman rank-correlation coefficients. | Proprietary stand-alone program using BMDP command language. | Class(es): L4a1b L4b1b L7a2a2 | Usage: @NBS*PLIB\$.BMDP BMDP3S | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP3S | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P3V** Uses maximum likelihood (ML) and restricted ML approaches to balanced and unbalanced fixed and random coefficient models of quite arbitrary form (including having covariates), with parameter estimation, hypothesis testing, and printing. Weights optional. | Proprietary stand-alone program using BMDP command language. | Class(es): L7a4 | Usage: @NBS*PLIB\$.BMDP BMDP3V | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP3V | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P4D** Counts frequency of each number, letter, or symbol in single-column fields (A1 format). Options: input case label variables in A4 format, diagnostic printing useful in preliminary data screening. Specified characters may be replaced by blanks or symbols. | Proprietary stand-alone program using BMDP command language. | Class(es): L1a1d N3 | Usage: @NBS*PLIB\$.BMDP BMDP4D | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP4D | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P4F** Forms two- or multi-way frequency tables. Options: percents; 25 tests and measures for two-way tables; fits and tests log-linear models, tests of marginal and partial association, stepwise models, three forms of input, structural zeros, cell and strata deletion, residuals. | Proprietary stand-alone program using BMDP command language. | Class(es): L9 | Usage: @NBS*PLIB\$.BMDP BMDP4F | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP4F | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P4M** Provides four methods of initial factor extraction from a correlation or covariance matrix, and several methods of rotation, prints shaded correlations, factor loadings, factor score coefficients, factor scores, Mahalanobis distances, and plots. Options: weights, form of input, save results. | Proprietary stand-alone program using BMDP command language. | Class(es): L13 | Usage: @NBS*PLIB\$.BMDP BMDP4M | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP4M | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P4R** Regression analysis for a dependent variable on a set of principal components computed from the independent variables in a stepwise manner determined either by magnitude of eigenvalue or correlations between dependent variable and components, with printing. Options: form of input, more printing, four plots. | Proprietary stand-alone program using BMDP command language. | Class(es): L8a4a1d | Usage: @NBS*PLIB\$.BMDP BMDP4R | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP4R | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P4V** Interactive or batch univariate and multivariate ANOVA and ANCOVA, including nested, repeated measures, split-plot, and changeover designs, and model building features. Options: cell weights for hypothesis testing, contrasts, tests of simple effects, save cell means. | Proprietary stand-alone program using BMDP command language. | Class(es): L7 | Usage: @NBS*PLIB\$.BMDP BMDP4V | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP4V | Access: @NBS*PLIB\$.BMDP BMD<program name>
- P5D** Prints histograms with frequencies and percentages, normal and detrended normal probability plots, halfnormal plots, cumulative frequency distribution plots, and cumulative histograms for ungrouped data or for grouped data – either separately or combined in one plot. Plot options. | Proprietary stand-alone program using BMDP command language. | Class(es): L3b L3c4h L3c4n | Usage: @NBS*PLIB\$.BMDP BMDP5D | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP5D | Access: @NBS*PLIB\$.BMDP BMD<program name>

- P5R** Least squares fit of a polynomial in one independent variable to the dependent variable. Prints standard results and goodness-of-fit statistics for each polynomial degree. Computations use orthogonal polynomials. Options: weights, additional printing, and three plots. | Proprietary stand-alone program using BMDP command language. | Class(es): L8a2b | Usage: @NBS*PLIB\$.BMDP BMDP5R | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP5R | Access: @NBS*PLIB\$.BMDP BMD <program name>
- P6D** Bivariate (scatter) plots. Options: several variables, or subsets of one variable (symbols identify group membership), on the same plot; prints correlation and linear regression statistics (line is marked on plot frame); user control for plot size, scales, and symbols. | Proprietary stand-alone program using BMDP command language. | Class(es): L3c | Usage: @NBS*PLIB\$.BMDP BMDP6D | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP6D | Access: @NBS*PLIB\$.BMDP BMD <program name>
- P6M** Computes canonical correlation analysis for two sets of variables and Bartlett's test for the significance of the remaining eigenvalues, with printing. Options: weights, form of input, additional printing and plotting, save results. | Proprietary stand-alone program using BMDP command language. | Class(es): L11 | Usage: @NBS*PLIB\$.BMDP BMDP6M | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP6M | Access: @NBS*PLIB\$.BMDP BMD <program name>
- P6R** Computes the partial correlations of a set of variables after removing the linear effects of a second set of variables. Can be used for regression, especially if multiple dependent variables are present. Prints standard results. Options: weights, form of input, additional printing and plots. | Proprietary stand-alone program using BMDP command language. | Class(es): L11 L8a7 | Usage: @NBS*PLIB\$.BMDP BMDP6R | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP6R | Access: @NBS*PLIB\$.BMDP BMD <program name>
- P7D** Side-by-side histograms for each cell in one-way or two-way ANOVA, within-group summary statistics and ANOVA table (with equality of variance test and tests that do not assume equal variances). Options: trimmed mean analysis, ANOVA diagnostics, tests of pairwise mean comparisons, correlations, Winsorized means. | Proprietary stand-alone program using BMDP command language. | Class(es): L7a1 L7a2a1a | Usage: @NBS*PLIB\$.BMDP BMDP7D | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP7D | Access: @NBS*PLIB\$.BMDP BMD <program name>
- P7M** Stepwise forward or backward discriminant analysis (including jackknifed classifications, percent correctly classified, Mahalanobis' distances, canonical variable coefficients, eigenvalues, scores, and plot of first two canonical variables). Options: interactive stepping, save results. | Proprietary stand-alone program using BMDP command language. | Class(es): L12 | Usage: @NBS*PLIB\$.BMDP BMDP7M | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP7M | Access: @NBS*PLIB\$.BMDP BMD <program name>
- P8D** Four methods to compute covariance and correlation matrices when data contain missing values or values out of range. Options: weights, summary statistics, save results, pairwise t-tests based on the pattern of incomplete data. | Proprietary stand-alone program using BMDP command language. | Class(es): L1e2 | Usage: @NBS*PLIB\$.BMDP BMDP8D | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP8D | Access: @NBS*PLIB\$.BMDP BMD <program name>
- P8M** Boolean factor analysis of binary (dichotomous) data. Options: initial estimates of the loading matrix, printing, save results. | Proprietary stand-alone program using BMDP command language. | Class(es): L13 | Usage: @NBS*PLIB\$.BMDP BMDP8M | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP8M | Access: @NBS*PLIB\$.BMDP BMD <program name>
- P8V** ANOVA for complete designs with equal cell sizes - nested, crossed, partially nested, partially crossed designs for fixed-effect models, mixed models (including repeated measures), and random-effect models, with parameter estimation and printing. | Proprietary stand-alone program using BMDP command language. | Class(es): L7a2a | Usage: @NBS*PLIB\$.BMDP BMDP8V | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP8V | Access: @NBS*PLIB\$.BMDP BMD <program name>
- P9D** Provides descriptive statistics (means, std. devs., frequencies, one-way ANOVA table) of groups (cells) for data classified into cells using one or more grouping variables. Options: miniplots of cell means (eight per page), plot frames are defined by combinations of levels of grouping variables. | Proprietary stand-alone program using BMDP command language. | Class(es): L7a1 L7a2a | Usage: @NBS*PLIB\$.BMDP BMDP9D | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP9D | Access: @NBS*PLIB\$.BMDP BMD <program name>
- P9M** Scoring based on preference pairs - for each observation construct score as linear combination of variables with coefficients based on expert preference, in stepwise manner. Options: printing, plots, compare results when analysis is repeated for different judges. | Proprietary stand-alone program using BMDP command language. | Class(es): L8a4a1e | Usage: @NBS*PLIB\$.BMDP BMDP9M | On-line doc: @PRT BMDP*DOC.SUMMARY | Access: @NBS*PLIB\$.BMDP BMD <program name>
- P9R** Estimates regression equations for "best" (by R-squared, adjusted R-squared, or Mallows' C(p) criterion) subset of predictor variables by Furnival-Wilson algorithm. Options: weights, form of input, Durbin-Watson statistic. Cook's distance and several types of residuals may be printed, plotted, or saved. | Proprietary stand-alone program using BMDP command language. | Class(es): L8a5 L8a9 | Usage: @NBS*PLIB\$.BMDP BMDP9R | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP9R | Access: @NBS*PLIB\$.BMDP BMD <program name>
- PACF** Computes and graphs partial autocorrelations of a time series and optionally saves results. | Command in MINITAB Proprietary interactive system. Class(es): L10c | Usage: PACF [up to K lags] for series in C [put into C] | On-line doc: HELP PACF (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

- PAM** Describes pattern of invalid values (missing or out of range) for multivariate data. Options: weights, grouping, estimates covariance and correlation matrices by one of three methods (including maximum likelihood), replace invalid values using means or one of several regression procedures, plots, save results. | Proprietary stand-alone program using BMDP command language. | Class(es): L1e2 | Usage: @NBS*PLIB\$.BMDP BMDPAM | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDPAM | Access: @NBS*PLIB\$.BMDP BMD<program name>
- PAR** Performs nonlinear regression using pseudo-Gauss-Newton algorithm. Derivatives are NOT specified. Options: weights, linear inequality constraints, maximum likelihood, functions of parameters, ridging, four plots, fitting models defined by differential equations. | Proprietary stand-alone program using BMDP command language. | Class(es): L8g1a L8g2a L8b | Usage: *NBS*PLIB\$.BMDP BMDPAR | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDPAR | Access: @NBS*PLIB\$.BMDP BMD<program name>
- PARCDF** Computes the cumulative distribution function value for the Pareto distribution with tail length parameter GAMMA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1p | Usage: CALL PARCDF(X,GAMMA,CDF) | On-line doc: CALL GAMSDOC PARCDF (or @PRT DATAPAC*DOC.PARCDF) | Access: LIB NBS*DAPAC
- PARPLT** Generates a Pareto probability plot with tail length parameter GAMMA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c4p | Usage: CALL PARPLT(X,N,GAMMA) | On-line doc: CALL GAMSDOC PARPLT (or @PRT DATAPAC*DOC.PARPLT) | Access: LIB NBS*DAPAC
- PARPPF** Computes the percent point function value for the Pareto distribution with tail length parameter GAMMA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2p | Usage: CALL PARPPF(P,GAMMA,PPF) | On-line doc: CALL GAMSDOC PARPPF (or @PRT DATAPAC*DOC.PARPPF) | Access: LIB NBS*DAPAC
- PARRAN** Generates a random sample of size N from the Pareto distribution with tail length parameter GAMMA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a16 | Usage: CALL PARRAN(N,GAMMA,ISTART,X) | On-line doc: CALL GAMSDOC PARRAN (or @PRT DATAPAC*DOC.PARRAN) | Access: LIB NBS*DAPAC
- PCHFD** Evaluates a piecewise cubic Hermite function and its first derivative at an array of points. | Portable single precision Fortran subprogram in PCHIP sublibrary of CMLIB library. | Class(es): E3 | Usage: CALL PCHFD(N,X,F,D,INCFD,SKIP,NE,XE,FE,DE,IERR) | On-line doc: CALL GAMSDOC PCHFD (or @PRT CMLIB*DOC.PCHFD/PCHIP) | Tests: CMLIB*TEST-SOURCE.\$Q/PCHIP | Access: LIB NBS*CMLIB
- PCHFE** Evaluates a piecewise cubic Hermite function at an array of points. | Portable single precision Fortran subprogram in PCHIP sublibrary of CMLIB library. | Class(es): E3 | Usage: CALL PCHFE(N,X,F,D,INCFD,SKIP,NE,XE,FE,IERR) | On-line doc: CALL GAMSDOC PCHFE (or @PRT CMLIB*DOC.PCHFE/PCHIP) | Tests: CMLIB*TEST-SOURCE.\$Q/PCHIP | Access: LIB NBS*CMLIB
- PCHIA** Evaluates the definite integral of a piecewise cubic Hermite function over an arbitrary interval. | Portable single precision Fortran subprogram in PCHIP sublibrary of CMLIB library. | Class(es): E3 | Usage: R = PCHIA(N,X,F,D,INCFD,SKIP,A,B,IERR) | On-line doc: CALL GAMSDOC PCHIA (or @PRT CMLIB*DOC.PCHIA/PCHIP) | Tests: CMLIB*TEST-SOURCE.\$Q/PCHIP | Access: LIB NBS*CMLIB
- PCHIC** Determines a piecewise monotone, piecewise cubic Hermite interpolant to given data. User control is available over boundary conditions and/or treatment of points where monotonicity switches direction. | Portable single precision Fortran subprogram in PCHIP sublibrary of CMLIB library. | Class(es): E1a | Usage: CALL PCHIC(IC,VC,SWITCH,N,X,F,D,INCFD,WK,NWK,IERR) | On-line doc: CALL GAMSDOC PCHIC (or @PRT CMLIB*DOC.PCHIC/PCHIP) | Tests: CMLIB*TEST-SOURCE.\$Q/PCHIP | Access: LIB NBS*CMLIB | See also: PCHFE for evaluation. See package documentation for other facilities.
- PCHID** Evaluates the definite integral of a piecewise cubic Hermite function over an interval whose endpoints are data points. | Portable single precision Fortran subprogram in PCHIP sublibrary of CMLIB library. | Class(es): E3 | Usage: R = PCHID(N,X,F,D,INCFD,SKIP,IA,IB,IERR) | On-line doc: CALL GAMSDOC PCHID (or @PRT CMLIB*DOC.PCHID/PCHIP) | Tests: CMLIB*TEST-SOURCE.\$Q/PCHIP | Access: LIB NBS*CMLIB
- PCHIM** Determines a monotone piecewise cubic Hermite interpolant to given data. Default boundary values are provided which are compatible with monotonicity. The interpolant will have an extremum at each point where monotonicity switches direction. | Portable single precision Fortran subprogram in PCHIP sublibrary of CMLIB library. | Class(es): E1a | Usage: CALL PCHIM(N,X,F,D,INFD,IERR) | On-line doc: CALL GAMSDOC PCHIM (or @PRT CMLIB*DOC.PCHIM/PCHIP) | Tests: CMLIB*TEST-SOURCE.\$Q/PCHIP | Access: LIB NBS*CMLIB | See also: PCHFE for evaluation. See package documentation for other facilities.
- PCHMC** Checks a cubic Hermite function for monotonicity. | Portable single precision Fortran subprogram in PCHIP sublibrary of CMLIB library. | Class(es): E3 | Usage: CALL PCHMC(N,X,F,D,INCFD,SKIP,ISMON,IERR) | On-line doc: CALL GAMSDOC PCHMC (or @PRT CMLIB*DOC.PCHMC/PCHIP) | Tests: CMLIB*TEST-SOURCE.\$Q/PCHIP | Access: LIB NBS*CMLIB
- PCHSP** Determines the cubic spline interpolant to given data. User has control over boundary conditions. | Portable single precision Fortran subprogram in PCHIP sublibrary of CMLIB library. | Class(es): E1a | Usage: CALL PCHSP(IC,VC,N,X,F,D,INCFD,WK,NWK,IERR) | On-line doc: CALL GAMSDOC PCHSP (or @PRT CMLIB*DOC.PCHSP/PCHIP) | Tests: CMLIB*TEST-SOURCE.\$Q/PCHIP | Access: LIB NBS*CMLIB | See also: PCHFE for evaluation. See package documentation for other facilities.
- PDECOL** Solves general nonlinear systems of initial-boundary-value problems in one space dimension with general boundary conditions. Spatial derivatives may be of at most second order. Uses method of lines based on collocation of B-spline basis functions. | Portable single precision Fortran subprogram in PDELIB library. | Class(es): I2a1a | Usage: CALL

- PDECOL**(T0,TOUT,DT,XBKPT,EPS,NINT,KORD,NCC,NPDE,MF,INDEX, WORK,IWORK) | On-line doc: CALL GAMSDOC PDECOL (or @PRT PDELIB*DOC.PDECOL) | Tests: PDELIB*TEST-SOURCE.PDECOL | Access: LIB NBS*PDELIB
- PDETWO** Solves general nonlinear systems of initial-boundary-value problems in two spatial dimensions with quasi-linear boundary conditions. Uses the method of lines based upon finite differences on a user-specified rectangular mesh. | Portable single precision Fortran subprogram in PDELIB library. | Class(es): L2a1b | Usage: CALL DRIVEP(NODE,T0,H,U1,TOUT,EPS,MF,INDEX,WORK,IWORK,X,Y) | On-line doc: CALL GAMSDOC PDETWO (or @PRT PDELIB*DOC.PDETWO) | Tests: PDELIB*TEST-SOURCE.PDETWO | Access: LIB NBS*PDELIB
- PFIT** Performs linear least squares regression analysis of a polynomial model. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L8a2b1 | Usage: CALL PFIT (Y, X, N, ID, SCRAT, NS, RES) | On-line doc: CALL GAMSDOC PFIT (or @PRT STATLIB*DOC.PFIT) | Tests: STATLIB*TEST.DEMO2 | Access: LIB NBS*STATLIB
- PFITS** Performs linear least squares regression analysis of a polynomial model, with computed results returned to the user. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L8a2b1 | Usage: CALL PFITS (Y, X, N, ID, SCRAT, NS, RES, COEF, PV, SDPV, SDRES, VCV, IVCV, NPRT) | On-line doc: CALL GAMSDOC PFITS (or @PRT STATLIB*DOC.PFITS) | Tests: STATLIB*TEST.DEMO2 | Access: LIB NBS*STATLIB
- PFITW** Performs weighted linear least squares regression analysis of a polynomial model. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L8a2b2 | Usage: CALL PFITW (Y, X, WT, N, ID, SCRAT, NS, RES) | On-line doc: CALL GAMSDOC PFITW (or @PRT STATLIB*DOC.PFITW) | Tests: STATLIB*TEST.DEMO2 | Access: LIB NBS*STATLIB
- PFITWS** Performs weighted linear least squares regression analysis of a polynomial model with computed results returned to the user. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L8a2b2 | Usage: CALL PFITWS (Y, X, WT, N, ID, SCRAT, NS, RES, COEF, PV, SDPV, SDRES, VCV, IVCV, NPRT) | On-line doc: CALL GAMSDOC PFITWS (or @PRT STATLIB*DOC.PFITWS) | Tests: STATLIB*TEST.DEMO2 | Access: LIB NBS*STATLIB
- PFQAD** Computes integral on (X1,X2) of product of function and the ID-th derivative of B-spline which is in piecewise polynomial representation. | Portable single precision Fortran subprogram in BSPLINE sublibrary of CMLIB library. Double precision version is DPFQAD. | Class(es): H2a2a1 E3 K6 | Usage: CALL PFQAD(F,LDC,C,XI,LX1,K,ID,X1,X2,TOL,QUAD,IERR) | On-line doc: CALL GAMSDOC PFQAD (or @PRT CMLIB*DOC.PFQAD/BSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/BSPLINE | Access: LIB NBS*CMLIB
- PKM** By k-means procedure, partitions a set of cases (observations) into k clusters - beginning with user-specified initial clusters or one cluster, proceeding in divisive stepwise manner, then doing iterative reallocation - prints cluster profile and plot. Options: weights, standardize data (four ways), save results. | Proprietary stand-alone program using BMDP command language. | Class(es): L14a2 | Usage: @NBS*PLIB\$.BMDP BMDPKM | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDPKM | Access: @NBS*PLIB\$.BMDP BMD<program name>
- PLOD** An easy to use interactive system for the solution of initial value problems for ordinary differential equations. Requires a Tektronix or Hewlett Packard graphics terminal. The user can change initial conditions, interval, parameters etc., and examine various plots on the terminal. Little programming needed. | Portable single precision Fortran subprogram in PLOD library. | Class(es): 11a | On-line doc: @PRT PLOD*DOC.SUMMARY | Access: @XQT NBS*PLOD.PLOD (or XQT NBS*PLOD.PLOD in CTS)
- PLOT** Yields a one-page printer plot of Y(I) versus X(I). | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c1 Q1 | Usage: CALL PLOT (Y,X,N) | On-line doc: CALL GAMSDOC PLOT (or @PRT DATAPAC*DOC.PLOT) | Access: LIB NBS*DAPAC
- PLOT** Prints a scatter diagram, with optional scale specification. | Command in MINITAB Proprietary interactive system. Class(es): L3c1 Q1 | Usage: PLOT y in C [from K to K] vs x in C [from K to K] | On-line doc: HELP PLOT (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- PLOT10** Yields a one-page printer plot of Y(I) versus X(I) for a subset of the data, with special plot characters, and with specified axis limits and labels. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c2 Q1 | Usage: CALL PLOT10(Y,X,CHAR,N,YMIN,YMAX,XMIN,XMAX,D,DMIN,DMAX,YAXID,XAXID,PLCHID) | On-line doc: CALL GAMSDOC PLOT10 (or @PRT DATAPAC*DOC.PLOT10) | Access: LIB NBS*DAPAC
- PLOT6** Yields a one-page printer plot of Y(I) versus X(I) for specified axis limits. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c1 Q1 | Usage: CALL PLOT6(Y,X,N,YMIN,YMAX,XMIN,XMAX) | On-line doc: CALL GAMSDOC PLOT6 (or @PRT DATAPAC*DOC.PLOT6) | Access: LIB NBS*DAPAC
- PLOT7** Yields a one-page printer plot of Y(I) versus X(I) with special plot characters and for specified axis limits. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c2 Q1 | Usage: CALL PLOT7(Y,X,CHAR,N,YMIN,YMAX,XMIN,XMAX) | On-line doc: CALL GAMSDOC PLOT7 (or @PRT DATAPAC*DOC.PLOT7) | Access: LIB NBS*DAPAC
- PLOT8** Yields a one-page printer plot of Y(I) versus X(I) with special plot characters for a subset of the data with specified axis limits. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c2 Q1 | Usage: CALL PLOT8(Y,X,CHAR,N,YMIN,YMAX,XMIN,XMAX,D,DMIN,DMAX) | On-line doc: CALL GAMSDOC PLOT8 (or @PRT DATAPAC*DOC.PLOT8) | Access: LIB NBS*DAPAC
- PLOT9** Yields a one-page printer plot of Y(I) versus X(I) with special plot characters and for specified axis limits and axis labels. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c2 Q1 | Usage: CALL

- PLOT9**(Y,X,CHAR,N,YMIN,YMAX,XMIN,XMAX,YAXID,XAXID,PLCHID) | On-line doc: CALL GAMSDOC PLOT9 (or @PRT DATAPAC*DOC.PLOT9) | Access: LIB NBS*DATAPAC
- PLOT C** Yields a one-page printer plot of Y(I) versus X(I) with special plotting characters. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c2 Q1 | Usage: CALL PLOT C(Y,X,CHAR,N) | On-line doc: CALL GAMSDOC PLOT C (or @PRT DATAPAC*DOC.PLOT C) | Access: LIB NBS*DATAPAC
- PLOTCT** Yields a narrow-width (71-character) plot of Y(I) versus X(I) with special plotting characters. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c2 Q1 | Usage: CALL PLOTCT(Y,X,CHAR,N) | On-line doc: CALL GAMSDOC PLOTCT (or @PRT DATAPAC*DOC.PLOTCT) | Access: LIB NBS*DATAPAC
- PLOTS** Yields a one-page printer plot of Y(I) versus X(I) for a subset of the data. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c1 Q1 | Usage: CALL PLOTS(Y,X,N,D,DMIN,DMAX) | On-line doc: CALL GAMSDOC PLOTS (or @PRT DATAPAC*DOC.PLOTS) | Access: LIB NBS*DATAPAC
- PLOTSC** Yields a one-page printer plot of Y(I) versus X(I) with special characters for a subset of the data. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c2 Q1 | Usage: CALL PLOTSC(Y,X,CHAR,N,D,DMIN,DMAX) | On-line doc: CALL GAMSDOC PLOTSC (or @PRT DATAPAC*DOC.PLOTSC) | Access: LIB NBS*DATAPAC
- PLOTST** Yields a narrow-width (71-character) of Y(I) versus X(I) for a subset of the data. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c1 Q1 | Usage: CALL PLOTST(Y,X,N,D,DMIN,DMAX) | On-line doc: CALL GAMSDOC PLOTST (or @PRT DATAPAC*DOC.PLOTST) | Access: LIB NBS*DATAPAC
- PLOTT** Yields a narrow-width (71-character) plot of Y(I) versus X(I). | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c1 Q1 | Usage: CALL PLOTT(Y,X,N) | On-line doc: CALL GAMSDOC PLOTT (or @PRT DATAPAC*DOC.PLOTT) | Access: LIB NBS*DATAPAC
- PLOTU** Produces 4 plots: data plot (X(I) versus I), autoregression plot (X(I) versus X(I-1)), histogram, and normal probability plot. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c5 L3a L3c4n | Usage: CALL PLOTU(X,N) | On-line doc: CALL GAMSDOC PLOTU (or @PRT DATAPAC*DOC.PLOTU) | Access: LIB NBS*DATAPAC
- PLOTX** Yields a one-page printer plot of X(I) versus I. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c5 Q1 | Usage: CALL PLOTX(X,N) | On-line doc: CALL GAMSDOC PLOTX (or @PRT DATAPAC*DOC.PLOTX) | Access: LIB NBS*DATAPAC
- PLOTXT** Yields a narrow-width (71-character) plot of X(I) versus I. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c5 Q1 | Usage: CALL PLOTXT(X,N) | On-line doc: CALL GAMSDOC PLOTXT (or @PRT DATAPAC*DOC.PLOTXT) | Access: LIB NBS*DATAPAC
- PLOTXX** Yields a one-page printer plot of X(I) versus X(I-1) for testing autocorrelation. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c5 | Usage: CALL PLOTXX(X,N) | On-line doc: CALL GAMSDOC PLOTXX (or @PRT DATAPAC*DOC.PLOTXX) | Access: LIB NBS*DATAPAC
- PLR** Stepwise logistic regression for binary dependent variable and categorical (design variables are formed) and continuous independent variables, using either maximum likelihood or approximate asymptotic estimates for stepping. Three options for generating design variables, plots, interactive stepping. | Proprietary stand-alone program using BMDP command language. | Class(es): L8a4d | Usage: @NBS*PLIB\$.BMDP BMDPLR | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDPLR | Access: @NBS*PLIB\$.BMDP BMD <program name>
- PLT** Displays a 50x100 character line printer scatter plot. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c1 Q1 | Usage: CALL PLT (Y, X, N) | On-line doc: CALL GAMSDOC PLT (or @PRT STATLIB*DOC.PLT) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB
- PLTH** Displays a 50x50 character line printer scatter plot. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c1 Q1 | Usage: CALL PLTH (Y, X, N) | On-line doc: CALL GAMSDOC PLTH (or @PRT STATLIB*DOC.PLTH) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB
- PLTHL** Displays a 50x50 character line printer scatter plot with user control of plot limits. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c1 Q1 | Usage: CALL PLTHL (Y, X, N, NOUT, YLB, YUB, XLB, XUB) | On-line doc: CALL GAMSDOC PLTHL (or @PRT STATLIB*DOC.PLTHL) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB
- PLTL** Displays a 50x100 character line printer scatter plot with user control of plot limits. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c1 Q1 | Usage: CALL PLTL (Y, X, N, NOUT, YLB, YUB, XLB, XUB) | On-line doc: CALL GAMSDOC PLTL (or @PRT STATLIB*DOC.PLTL) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB
- PLTSCT** Yields a narrow-width (71-character) plot of Y(I) versus X(I) with special plot characters and a subset of the data. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c2 Q1 | Usage: CALL PLTSCT(Y,X,CHAR,N,D,DMIN,DMAX) | On-line doc: CALL GAMSDOC PLTSCT (or @PRT DATAPAC*DOC.PLTSCT) | Access: LIB NBS*DATAPAC
- PLTXXT** Yields a narrow-width (71-character) plot of X(I) versus X(I-1) for testing autocorrelation. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c5 | Usage: CALL PLTXXT(X,N) | On-line doc: CALL GAMSDOC PLTXXT (or @PRT

DATA PAC*DOC.PLTXXT) | Access: LIB NBS*DATA PAC

POCH Pochhammer's symbol $(a)_{\text{sub } x} = \Gamma(a+x) / \Gamma(a)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DPOCH. | Class(es): C1 C7a | Usage: $Y = \text{POCH}(A,X)$ | On-line doc: CALL GAMSDOC POCH (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

POCH1 Pochhammer's symbol from first order, $= ((a)_{\text{sub } x - 1}) / x$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DPOCH1. | Class(es): C1 C7a | Usage: $Y = \text{POCH1}(A,X)$ | On-line doc: CALL GAMSDOC POCH1 (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

POICDF Computes the cumulative distribution function value at X for the Poisson distribution with tail length parameter ALAMBA. | Portable single precision Fortran subprogram in DATA PAC library. | Class(es): L5a1p | Usage: CALL POICDF(X,ALAMBA,CDF) | On-line doc: CALL GAMSDOC POICDF (or @PRT DATA PAC*DOC.POICDF) | Access: LIB NBS*DATA PAC

POIPLT Generates a Poisson probability plot with tail length parameter ALAMBA, mean = ALAMBA and standard deviation = $\sqrt{\text{ALAMBA}}$. | Portable single precision Fortran subprogram in DATA PAC library. | Class(es): L3c4p | Usage: CALL POIPLT(X,N,ALAMBA) | On-line doc: CALL GAMSDOC POIPLT (or @PRT DATA PAC*DOC.POIPLT) | Access: LIB NBS*DATA PAC

POIPPF Computes the percent point function value at P for the Poisson distribution with mean = ALAMBA and standard deviation = $\sqrt{\text{ALAMBA}}$. | Portable single precision Fortran subprogram in DATA PAC library. | Class(es): L5a2p | Usage: CALL POIPPF(P,ALAMBA,PPF) | On-line doc: CALL GAMSDOC POIPPF (or @PRT DATA PAC*DOC.POIPPF) | Access: LIB NBS*DATA PAC

POIRAN Generates a random sample of size N from the Poisson distribution with mean = ALAMBA and standard deviation = $\sqrt{\text{ALAMBA}}$. | Portable single precision Fortran subprogram in DATA PAC library. | Class(es): L6a16 | Usage: CALL POIRAN(N,ALAMBA,ISTART,X) | On-line doc: CALL GAMSDOC POIRAN (or @PRT DATA PAC*DOC.POIRAN) | Access: LIB NBS*DATA PAC

POIS3D Solves block tridiagonal linear systems of algebraic equations arising from the discretization of separable elliptic partial differential equations in 3D. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): D2a4 I2b4b | Usage: CALL POIS3D(MPEROD,L,C1,MPEROD,M,C2,NPEROD,N,A,B,D,LDIMF,MDIMF, F,IERROR,W) | On-line doc: CALL GAMSDOC POIS3D (or @PRT CMLIB*DOC.POIS3D/FSHPK) | Tests: CMLIB*TEST-SOURCE.POIS3D/FSHPK | Access: LIB NBS*CMLIB

POISSON Prints table of Poisson probabilities and cumulative distribution function. | Command in MINITAB Proprietary interactive system. | Class(es): L5a1p | Usage: POISSON probabilities for mean K | On-line doc: HELP POISSON (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

POISTG Solves block tridiagonal linear systems of algebraic equations arising from the discretization of separable elliptic partial differential equations. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): D2a4 I2b4b | Usage: CALL POISTG(NPEROD,N,MPEROD,M,A,B,C,IDIMY,Y,IERROR,W) | On-line doc: CALL GAMSDOC POISTG (or @PRT CMLIB*DOC.POISTG/FSHPK) | Tests: CMLIB*TEST-SOURCE.POISTG/FSHPK | Access: LIB NBS*CMLIB

POLY Computes a least squares polynomial fit (of degree = IDEG) of the response variable data in the vector Y as a function of vector X and with optional weights. | Portable single precision Fortran subprogram in DATA PAC library. | Class(es): L8a2a | Usage: CALL POLY(Y,X,W,N,IDEG,IWRITE,B,SDB,S,DF,PRED,RES) | On-line doc: CALL GAMSDOC POLY (or @PRT DATA PAC*DOC.POLY) | Access: LIB NBS*DATA PAC

PPQAD Computes the integral of a B-spline from X1 to X2. The B-spline must be in piecewise polynomial representation. | Portable single precision Fortran subprogram in BSPLINE sublibrary of CMLIB library. Double precision version is DPPQAD. | Class(es): H2a2a1 E3 K6 | Usage: CALL PPQAD(LDC,C,X1,LX1,K,X1,X2,PQUAD) | On-line doc: CALL GAMSDOC PPQAD (or @PRT CMLIB*DOC.PPQAD/BSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/BSPLINE | Access: LIB NBS*CMLIB

PPVAL Calculates (at X) the value of the IDERIV-th derivative of the B-spline from its piecewise polynomial representation. | Portable single precision Fortran subprogram in BSPLINE sublibrary of CMLIB library. Double precision version is DPPVAL. | Class(es): E3 K6 | Usage: CALL PPVAL(LDC,C,X1,LX1,K,IDERIV,X,INPPV) | On-line doc: CALL GAMSDOC PPVAL (or @PRT CMLIB*DOC.PPVAL/BSPLINE and CMLIB*DOC.SUMMARY/BSPLINE) | Tests: CMLIB*TEST-SOURCE.\$F/BSPLINE | Access: LIB NBS*CMLIB

PRANDOM Generates K pseudo-random numbers from the Poisson distribution with specified population mean K. | Command in MINITAB Proprietary interactive system. | Class(es): L6a16 | Usage: PRANDOM K Poisson observations with population mean = K, put into C | On-line doc: HELP PRANDOM (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

PROPOR Computes the sample proportion which is the proportion of data between XMIN and XMAX (inclusively) in the input vector X. | Portable single precision Fortran subprogram in DATA PAC library. | Class(es): L1a1d | Usage: CALL PROPOR(X,N,XMIN,XMAX,IWRITE,XPROP) | On-line doc: CALL GAMSDOC PROPOR (or @PRT DATA PAC*DOC.PROPOR) | Access: LIB NBS*DATA PAC

PSI Psi (digamma), $= \gamma'(x) / \gamma(x)$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DPSI. | Class(es): C7c | Usage: $Y = \text{PSI}(X)$ | On-line doc: CALL GAMSDOC PSI (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

PUMB Given a basic mesh, this subdivides each interval into a uniform but variable number of points. Multiplicities can occur. | Proprietary

single precision Fortran subprogram in PORT library. Double precision version is DPUMB. | Class(es): E3 K6 | Usage: CALL PUMB (XB,NXB,NA,K,X,NX) | On-line doc: CALL GAMSDOC PUMB (or @PRT PORT*DOC.PUMB) | Access: LIB NBS*PORT

PUMD Given a basic mesh, this subdivides each interval into a uniform but variable number of points. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DPUMD. | Class(es): E3 K6 | Usage: CALL PUMD (XB,NXB,NA,X,NX) | On-line doc: CALL GAMSDOC PUMD (or @PRT PORT*DOC.PUMD) | Access: LIB NBS*PORT

Q

- Q1DA** Automatic evaluation of a user-defined function of one variable. Special features include randomization and singularity weakening. | Portable single precision Fortran subprogram in Q1DA sublibrary of CMLIB library. | Class(es): H2a1a1 | Usage: CALL Q1DA(A,B,EPS,R,E,KF,IFLAG) | On-line doc: CALL GAMSDOC Q1DA (or @PRT CMLIB*DOC.Q1DA/Q1DA) | Tests: CMLIB*TEST-SOURCE.\$Q/Q1DA | Access: LIB NBS*CMLIB
- Q1DAX** Flexible subroutine for the automatic evaluation of definite integrals of a user-defined function of one variable. Special features include randomization, singularity weakening, restarting, specification of an initial mesh (optional), and output of smallest and largest integrand values. | Portable single precision Fortran subprogram in Q1DA sublibrary of CMLIB library. | Class(es): H2a1a1 | Usage: CALL Q1DAX(F,A,B,EPS,R,E,NINT,RST,W,NMAX,FMIX,FMAX,KF,IFLAG) | On-line doc: CALL GAMSDOC Q1DAX (or @PRT CMLIB*DOC.Q1DAX/Q1DA) | Tests: CMLIB*TEST-SOURCE.\$Q/Q1DA | Access: LIB NBS*CMLIB
- Q1DB** Automatic evaluation of a user-defined function of one variable. Integrand must be a Fortran Function but user may select name. Special features include randomization and singularity weakening. Intermediate in usage difficulty between Q1DA and Q1DAX. | Portable single precision Fortran subprogram in Q1DA sublibrary of CMLIB library. | Class(es): H2a1a1 | Usage: CALL Q1DB(F,A,B,EPS,RESULT,ERREST,NFUNC,IFLAG) | On-line doc: CALL GAMSDOC Q1DB (or @PRT CMLIB*DOC.Q1DB/Q1DA) | Tests: CMLIB*TEST-SOURCE.\$Q/Q1DA | Access: LIB NBS*CMLIB
- QAG** Automatic adaptive integrator, will handle many non-smooth integrands using Gauss Kronrod formulas. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAG. | Class(es): H2a1a1 | Usage: CALL QAG(F,A,B,EPSABS,EPSREL,KEY,RESULT,ABSERR,NEVAL,IER,LIMIT, LENW, LAST, IWORK, WORK) | On-line doc: CALL GAMSDOC QAG (or @PRT CMLIB*DOC.QAG/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.QAG/QUADSP, CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QAGE** Automatic adaptive integrator, can handle most non-smooth functions also provides more information than QAG. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAGE. | Class(es): H2a1a1 | Usage: CALL QAGE(F,A,B,EPSABS,EPSREL,KEY,LIMIT,RESULT,ABSERR,NEVAL, IER,ALIST, BLIST, RLIST,ELIST,IORD, LAST) | On-line doc: CALL GAMSDOC QAGE (or @PRT CMLIB*DOC.QAGE/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QAGI** Automatic adaptive integrator for semi-infinite or infinite intervals. Uses nonlinear transformation and extrapolation. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAGI. | Class(es): H2a3a1 H2a4a1 | Usage: CALL QAGI(F,BOUND,INF,EPSABS,EPSREL,RESULT,ABSERR,NEVAL,IER, LIMIT, LENW, LAST, IWORK, WORK) | On-line doc: CALL GAMSDOC QAGI (or @PRT CMLIB*DOC.QAGI/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.QAGI/QUADSP, CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QAGIE** Automatic integrator for semi-infinite or infinite intervals and general integrands, provides more information than QAGI. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAGIE. | Class(es): H2a3a1 H2a4a1 | Usage: CALL QAGIE(F,BOUND,INF,EPSABS,EPSREL,LIMIT,RESULT,ABSERR,NEVAL, IER,ALIST, BLIST, RLIST,ELIST,IORD, LAST) | On-line doc: CALL GAMSDOC QAGIE (or @PRT CMLIB*DOC.QAGIE/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QAGP** Automatic adaptive integrator, allows user to specify location of singularities or difficulties of integrand, uses extrapolation. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAGP. | Class(es): H2a2a1 | Usage: CALL QAGP(F,A,B,NPTS2,POINTS,EPSABS,EPSREL,RESULT,ABSERR,NEVAL, IER, LENIW, LENW, IWORK, WORK) | On-line doc: CALL GAMSDOC QAGP (or @PRT CMLIB*DOC.QAGP/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.QAGP/QUADSP, CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QAGPE** Automatic adaptive integrator for function with user specified endpoint singularities, provides more information than QAGP. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAGPE. | Class(es): H2a2a1 | Usage: CALL QAGPE(F,A,B,NPTS2,POINTS,EPSABS,EPSREL,LIMIT,RESULT, ABSERR,NEVAL,IER, ALIST BLIST,RLIST,ELIST,PTS,IORD,LEVEL,NDIN, LAST) | On-line doc: CALL GAMSDOC QAGPE (or @PRT CMLIB*DOC.QAGPE/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QAGS** Automatic adaptive integrator, will handle most non-smooth integrands including those with endpoint singularities, uses extrapolation. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAGS. | Class(es): H2a1a1 | Usage: CALL QAGS(F,A,B,EPSABS,EPSREL,RESULT,ABSERR,NEVAL,IER,LIMIT, LENW, LAST,IWORK, WORK) | On-line doc: CALL GAMSDOC QAGS (or @PRT CMLIB*DOC.QAGS/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.QAGS/QUADSP, CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QAGSE** Automatic adaptive integrator, can handle intergands with endpoint singularities provides more information than QAGS. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAGSE. | Class(es): H2a1a1 | Usage: CALL QAGSE(F,A,B,EPSABS,EPSREL,LIMIT,RESULT,ABSERR,NEVAL,IER, ALIST, BLIST,RLIST, ELIST,IORD, LAST) | On-line doc: CALL GAMSDOC QAGSE (or @PRT CMLIB*DOC.QAGSE/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB

- QAWC** Cauchy principal value integrator, using adaptive Clenshaw Curtis method (real Hilbert transform). | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAWC. | Class(es): H2a2a1 J4 | Usage: CALL QAWC(F,A,B,C, EPSABS, EPSREL, RESULT, ABSERR, NEVAL, IER, LIMIT, LENW, LAST, IWORK, WORK) | On-line doc: CALL GAMSDOC QAWC (or @PRT CMLIB*DOC.QAWC/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.QAWC/QUADSP, CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QAWCE** Cauchy Principal value integrator, provides more information than QAWC (real Hilbert transform). | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAWCE. | Class(es): H2a2a1 J4 | Usage: CALL QAWCE(F,A,B,C, EPSABS, EPSREL, LIMIT, RESULT, ABSERR, NEVAL, IER, ALIST, BLIST, RLIST, ELIST, IORD, LAST) | On-line doc: CALL GAMSDOC QAWCE (or @PRT CMLIB*DOC.QAWCE/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QAWF** Automatic integrator for Fourier integrals on (a, infinity) with factors SIN(OMEGA*X), COS(OMEGA*X) by integrating between zeros. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAWF. | Class(es): H2a3a1 | Usage: CALL QAWF(F,A,OMEGA, INTEGR, EPSABS, RESULT, ABSERR, NEVAL, IER, LIMLST, LST, LENIW, MAXP1, LENW, IWORK, WORK) | On-line doc: CALL GAMSDOC QAWF (or @PRT CMLIB*DOC.QAWF/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.QAWF/QUADSP, CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QAWFE** Automatic integrator for Fourier integrals, with SIN(OMEGA*X) factor on (A, INFINITY), provides more information than QAWF. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAWFE. | Class(es): H2a3a1 | Usage: CALL QAWFE(F,A,OMEGA, INTEGR, EPSABS, LIMLST, LIMIT, MAXP1, RESULT, ABSERR, NEVAL, IER, RSLST, ERLST, IERLST, LST, ALIST, BLIST, RLIST, ELIST, IORD, NNLOG, CHEBMO) | On-line doc: CALL GAMSDOC QAWFE (or @PRT CMLIB*DOC.QAWFE/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QAWO** Automatic adaptive integrator for integrands with oscillatory sin or cosine factor. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAWO. | Class(es): H2a2a1 | Usage: CALL QAWO(F,A,B,OMEGA, INTEGR, EPSABS, EPSREL, RESULT, ABSERR, NEVAL, IER, LENIW, MAXP1, LENW, LAST, IWORK, WORK) | On-line doc: CALL GAMSDOC QAWO (or @PRT CMLIB*DOC.QAWO/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.QAWO/QUADSP, CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QAWOE** Automatic integrator for integrands with explicit oscillatory sin or cosine factor, provides more information than QAWO. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAWOE. | Class(es): H2a2a1 | Usage: CALL QAWOE(F,A,B,OMEGA, INTEGR, EPSABS, EPSREL, LIMIT, ICALL, MAXP1, RESULT, ABSERR, NEVAL, IER, ALIST, BLIST, RLIST, ELIST, IORD, NNLOG, MOMCOM, CHEBMO) | On-line doc: CALL GAMSDOC QAWOE (or @PRT CMLIB*DOC.QAWOE/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QAWS** Automatic integrator for functions with explicit algebraic and/or logarithmic endpoint singularities. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAWS. | Class(es): H2a2a1 | Usage: CALL QAWS(F,A,B,ALFA, BETA, INTEGR, EPSABS, EPSREL, RESULT, ABSERR, NEVAL, IER, LIMIT, LENW, LAST, IWORK, WORK) | On-line doc: CALL GAMSDOC QAWS (or @PRT CMLIB*DOC.QAWS/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.QAWS/QUADSP, CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QAWSE** Automatic integrator for integrands with explicit algebraic and/or logarithmic endpoint singularities, more information than QAWS. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAWSE. | Class(es): H2a2a1 | Usage: CALL QAWSE(F,A,B,ALFA, BETA, INTEGR, EPSABS, EPSREL, LIMIT, RESULT, ABSERR, NEVAL, IER, ALIST, BLIST, RLIST, ELIST, IORD, LAST) | On-line doc: CALL GAMSDOC QAWSE (or @PRT CMLIB*DOC.QAWSE/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QC25C** Uses 25 point Clenshaw-Curtis formula to estimate integral of F*W where $W=1/(X-C)$. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQC25C. | Class(es): H2a2a2 J4 | Usage: CALL QC25C(F,A,B,C, RESULT, ABSERR, KRUL, NEVAL) | On-line doc: CALL GAMSDOC QC25C (or @PRT CMLIB*DOC.QC25C/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QC25F** Clenshaw-Curtis integration rule for function with cos or sin factor, also uses Gauss Kronrod formula. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQC25F. | Class(es): H2a2a2 | Usage: CALL QC25F(F,A,B,OMEGA, INTEGR, NRMOM, MAXP1, KSAVE, RESULT, ABSERR, NEVAL, RESABS, RESASC, MOMCOM, CHEBMO) | On-line doc: CALL GAMSDOC QC25F (or @PRT CMLIB*DOC.QC25F/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QC25S** Estimates integral of function with algebraico-logarithmic singularities with 25 point Clenshaw-Curtis formula and gives error estimate. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQC25S. | Class(es): H2a2a2 | Usage: CALL QC25S(F,A,B,BL,ALFA, BETA, RI, RJ, RG, RH, RESULT, ABSERR, RESASC) INTEGR, NEV) | On-line doc: CALL GAMSDOC QC25S (or @PRT CMLIB*DOC.QC25S/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB

- QK15** Evaluates integral of given function on an interval with a 15 point Gauss Kronrod formula and returns error estimate. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQK15. | Class(es): H2a1a2 | Usage: CALL QK15(F,A,B,RESULT,ABSERR,RESABS,RESASC) | On-line doc: CALL GAMSDOC QK15 (or @PRT CMLIB*DOC.QK15/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QK151** Evaluates integral of given function on semi-infinite or infinite interval with a transformed 15 point Gauss Kronrod formula and gives error ets. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQK151. | Class(es): H2a3a2 H2a4a2 | Usage: CALL QK151(F,BOUN,INF,A,B,RESULT,ABSERR,RESABS,RESASC) | On-line doc: CALL GAMSDOC QK151 (or @PRT CMLIB*DOC.QK151/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QK15W** Evaluates integral of given function times arbitrary weight function on interval with 15 point Gauss Kronrod formula and gives error estimate. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQK15W. | Class(es): H2a2a2 | Usage: CALL QK15W(F,W,P1,P2,P3,P4,KP,A,B,RESULT,ABSERR,RESABS,RESASC) | On-line doc: CALL GAMSDOC QK15W (or @PRT CMLIB*DOC.QK15W/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QK21** Evaluates integral of given function on an interval with a 21 point Gauss Kronrod formula and returns error estimate. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQK21. | Class(es): H2a1a2 | Usage: CALL QK21(F,A,B,RESULT,ABSERR,RESABS,RESASC) | On-line doc: CALL GAMSDOC QK21 (or @PRT CMLIB*DOC.QK21/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QK31** Evaluates integral of given function on an interval with a 31 point Gauss Kronrod formula and returns error estimate. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQK31. | Class(es): H2a1a2 | Usage: CALL QK31(F,A,B,RESULT,ABSERR,RESABS,RESASC) | On-line doc: CALL GAMSDOC QK31 (or @PRT CMLIB*DOC.QK31/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QK41** Evaluates integral of given function on an interval with a 41 point Gauss Kronrod formula and returns error estimate. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQK41. | Class(es): H2a1a2 | Usage: CALL QK41(F,A,B,RESULT,ABSERR,RESABS,RESASC) | On-line doc: CALL GAMSDOC QK41 (or @PRT CMLIB*DOC.QK41/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QK51** Evaluates integral of given function on an interval with a 51 point Gauss Kronrod formula and returns error estimate. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQK51. | Class(es): H2a1a2 | Usage: CALL QK51(F,A,B,RESULT,ABSERR,RESABS,RESASC) | On-line doc: CALL GAMSDOC QK51 (or @PRT CMLIB*DOC.QK51/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QK61** Evaluates integral of given function on an interval with a 61 point Gauss Kronrod formula and returns error estimate. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQK61. | Class(es): H2a1a2 | Usage: CALL QK61(F,A,B,RESULT,ABSERR,RESABS,RESASC) | On-line doc: CALL GAMSDOC QK61 (or @PRT CMLIB*DOC.QK61/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QMOMO** Computes integral of k-th degree Tchebycheff polynomial times selection of functions with various singularities. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQMOMO. | Class(es): H2a2a1 C3a2 | Usage: CALL QMOMO(ALFA,BETA,R1,RJ,RG,RH,INTEGR) | On-line doc: CALL GAMSDOC QMOMO (or @PRT CMLIB*DOC.QMOMO/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QNG** Automatic non-adaptive integrator for smooth functions, using Gauss Kronrod Patterson formulas. | Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQNG. | Class(es): H2a1a1 | Usage: CALL QNG(F,A,B,EPSABS,EPSREL,RESULT,ABSERR,NEVAL,IER) | On-line doc: CALL GAMSDOC QNG (or @PRT CMLIB*DOC.QNG/QUADSP and CMLIB*DOC.SUMMARY/QUADSP) | Tests: CMLIB*TEST-SOURCE.QNG/QUADSP, CMLIB*TEST-SOURCE.\$Q/QUADSP | Access: LIB NBS*CMLIB
- QUAD** Finds the integral of a general user defined EXTERNAL function by an adaptive technique to given absolute accuracy. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DQUAD. | Class(es): H2a1a1 | Usage: CALL QUAD(F,A,B,EPS,ANS,ERREST) | On-line doc: CALL GAMSDOC QUAD (or @PRT PORT*DOC.QUAD) | Access: LIB NBS*PORT
- QZHES** The first step of the QZ algorithm for solving generalized matrix eigenproblems. Accepts a pair of real general matrices and reduces one of them to upper Hessenberg form and the other to upper triangular form using orthogonal transformations. Usually followed by QZIT, QZVAL, QZVEC. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1b3 | Usage: CALL QZHES(NM,N,A,B,MATZ,Z) | On-line doc: CALL GAMSDOC QZHES (or @PRT CMLIB*DOC.QZHES/EISPACK) | Access: LIB NBS*CMLIB | See also: QZIT QZVAL QZVEC

- QZIT** The second step of the QZ algorithm for generalized eigenproblems. Accepts an upper Hessenberg and an upper triangular matrix and reduces the former to quasi-triangular form while preserving the form of the latter. Usually preceded by QZHES and followed by QZVAL and QZVEC. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1b3 | Usage: CALL QZIT(NM,N,A,B,EPS1,MATZ,Z,IERR) | On-line doc: CALL GAMSDOC QZIT (or @PRT CMLIB*DOC.QZIT/EISPACK) | Access: LIB NBS*CMLIB | See also: QZHES QZVAL QZVEC
- QZVAL** The third step of the QZ algorithm for generalized eigenproblems. Accepts a pair of real matrices, one in quasi-triangular form and the other in upper triangular form and computes the eigenvalues of the associated eigenproblem. Usually preceded by QZHES, QZIT, and followed by QZVEC. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c2c | Usage: CALL QZVAL(NM,N,A,B,ALFR,ALF1,BETA,MATZ,Z) | On-line doc: CALL GAMSDOC QZVAL (or @PRT CMLIB*DOC.QZVAL/EISPACK) | Access: LIB NBS*CMLIB | See also: QZHES QZIT QZVEC
- QZVEC** The optional fourth step of the QZ algorithm for generalized eigenproblems. Accepts a matrix in quasi-triangular form and another in upper triangular form and computes the eigenvectors of the triangular problem and transforms them back to the original coordinates. Usually preceded by QZHES, QZIT, QZVAL. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c3 | Usage: CALL QZVEC(NM,N,A,B,ALFR,ALF1,BETA,Z) | On-line doc: CALL GAMSDOC QZVEC (or @PRT CMLIB*DOC.QZVEC/EISPACK) | Access: LIB NBS*CMLIB | See also: QZHES QZIT QZVAL

R

- R1MACH** Provides single precision machine dependent information, e.g. R1MACH(4) returns machine epsilon. | Portable single precision Fortran subprogram in MACHCONST sublibrary of CMLIB library. Double precision version is D1MACH. | Class(es): R1 | Usage: R=R1MACH(I) | On-line doc: CALL GAMSDOC R1MACH (or @PRT CMLIB*DOC.R1MACH/MACHCONST) | Tests: CMLIB*TEST-SOURCE.\$Q/MACHCONST | Access: LIB NBS*CMLIB
- R1MACH** Provides the single precision machine-dependent constants required to adapt PORT library programs to individual computers. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is D1MACH. | Class(es): R1 | Usage: X = R1MACII (I) | On-line doc: CALL GAMSDOC R1MACH (or @PRT PORT*DOC.R1MACH) | Access: LIB NBS*PORT
- RANBYT** Returns the real random variate generated by UNI, together with its bit pattern presented in four 8-bit bytes. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): L6a21 | Usage: CALL RANBYT (UNI,IBYTE) | On-line doc: CALL GAMSDOC RANBYT (or @PRT PORT*DOC.RANBYT) | Access: LIB NBS*PORT | See also: UNI,RANSET
- RAND** Uniform random number on [0,1]. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): L6a21 | Usage: Y = RAND (X) | On-line doc: CALL GAMSDOC RAND (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- RANGE** Computes the sample range of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1b | Usage: CALL RANGE (X,N,IWRITE,XRANGE) | On-line doc: CALL GAMSDOC RANGE (or @PRT DATAPAC*DOC.RANGE) | Access: LIB NBS*DATAPAC
- RANK** Ranks (in ascending order) the N elements of the single precision vector X, and puts the resulting N ranks into the vector XR. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): N6a1b1 | Usage: CALL RANK(X,N,XR) | On-line doc: CALL GAMSDOC RANK (or @PRT DATAPAC*DOC.RANK) | Access: LIB NBS*DATAPAC
- RANK** Ranks the values in a vector. Ties are assigned the average rank. | Command in MINITAB Proprietary interactive system. Class(es): N6a1b1 | Usage: RANK the values in C, put ranks into C | On-line doc: HELP RANK (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- RANPER** Generates a random permutation of size N of the values 1.0, 2.0, 3.0, ..., N-1, N. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a16 | Usage: CALL RANPER(N,ISTART,X) | On-line doc: CALL GAMSDOC RANPER (or @PRT DATAPAC*DOC.RANPER) | Access: LIB NBS*DATAPAC
- RANSET** Initializes the uniform random number generator, UNI, to other than the default initial values. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): L6c | Usage: CALL RANSET (ICSEED,ITSEED) | On-line doc: CALL GAMSDOC RANSET (or @PRT PORT*DOC.RANSET) | Access: LIB NBS*PORT | See also: UNI,RANBYT
- RATQR** Computes largest or smallest eigenvalues of symmetric tridiagonal matrix using rational QR method with Newton correction. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 D4c2a | Usage: CALL RATQR(N,EPS1,D,E,E2,M,W,IND,BD,TYPE,IDEF,IERR) | On-line doc: CALL GAMSDOC RATQR (or @PRT CMLIB*DOC.RATQR/EISPACK) | Access: LIB NBS*CMLIB
- READ** Performs a format-free read. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): N1 | Usage: CALL READ(ICOL1,ICOL2,X,N) | On-line doc: CALL GAMSDOC READ (or @PRT DATAPAC*DOC.READ) | Access: LIB NBS*DATAPAC
- READG** Performs a format-free read of data from input unit = IRD. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): N1 | Usage: CALL READG(IRD,ICOL1,ICOL2,X,N) | On-line doc: CALL GAMSDOC READG (or @PRT DATAPAC*DOC.READG) | Access: LIB NBS*DATAPAC
- REBAK** Forms eigenvectors of generalized symmetric eigensystem from eigenvectors of derived matrix output from REDUC or REDUC2. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL REBAK(NM,N,B,DL,M,Z) | On-line doc: CALL GAMSDOC REBAK (or @PRT CMLIB*DOC.REBAK/EISPACK) | Access: LIB NBS*CMLIB | See also: REDUC REDUC2
- REBAKB** Forms eigenvectors of generalized symmetric eigensystem from eigenvectors of derived matrix output from REDUC2. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL REBAKB(NM,N,B,DL,M,Z) | On-line doc: CALL GAMSDOC REBAKB (or @PRT CMLIB*DOC.REBAKB/EISPACK) | Access: LIB NBS*CMLIB | See also: REDUC2
- REDUC** Reduces generalized symmetric eigenproblem $A \cdot X = (\text{LAMBDA}) \cdot B \cdot X$, to standard symmetric eigenproblem, using Cholesky factorization. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1c | Usage: CALL REDUC(NM,N,A,B,DL,IERR) | On-line doc: CALL GAMSDOC REDUC (or @PRT CMLIB*DOC.REDUC/EISPACK) | Access: LIB NBS*CMLIB
- REDUC2** Reduces certain generalized symmetric eigenproblems to standard symmetric eigenproblem, using Cholesky factorization. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1c | Usage: CALL REDUC2(NM,N,A,B,DL,IERR) | On-line doc: CALL GAMSDOC REDUC2 (or @PRT CMLIB*DOC.REDUC2/EISPACK) | Access: LIB NBS*CMLIB

- REGRESS** Performs simple or multiple linear regression, prints standard results. Options: amount of output, save results, weights, handle missing values, through the origin, compute and save regression diagnostics, lack of fit tests. | Command in MINITAB Proprietary interactive system. Class(es): L8a L8a9 | Usage: REGRESS C on K predictors C₁,...,C_K [put standardized residuals in C [fits in C]] ; subcommands NOCONSTANT; WEIGHTS in C; MSE into K; COEF into C; XPXINV into M; RMATRIX into M; HI into C; RESIDS into C; TRESIDS into C; COOKD into C; DFITS into C; VIF; PURE error lack of fit test; XLOF experimental lack of fit. | On-line doc: HELP REGRESS (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- RELS** Computes the sample relative standard deviation of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1b | Usage: CALL RELS(X,N,IWRITE,XRELS) | On-line doc: CALL GAMSDOC RELS (or @PRT DATAPAC*DOC.RELS) | Access: LIB NBS*DAPAC
- REPLAC** Replaces (with the value XNEW) all observations in the vector X which are inside the interval [XMIN, XMAX]. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L2a | Usage: CALL REPLAC(X,N,XMIN,XMAX,XNEW) | On-line doc: CALL GAMSDOC REPLAC (or @PRT DATAPAC*DOC.REPLAC) | Access: LIB NBS*DAPAC
- RETAIN** Retains all observations in the vector X which are inside the interval [XMIN, XMAX]. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L2d | Usage: CALL RETAIN(X,N,XMIN,XMAX,NEWN) | On-line doc: CALL GAMSDOC RETAIN (or @PRT DATAPAC*DOC.RETAIN) | Access: LIB NBS*DAPAC
- RETSRC** Test and reset error recovery mode for PORT library programs. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): R3a | Usage: CALL RETSRC (IOLD) | On-line doc: CALL GAMSDOC RETSRC (or @PRT PORT*DOC.RETSRC) | Access: LIB NBS*PORT
- RFFTB** Computes real periodic sequence from real Fourier coefficients. Performs Fourier synthesis. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): J1a1 | Usage: CALL RFFTB(N,R,WSAVE) | On-line doc: CALL GAMSDOC RFFTB (or @PRT CMLIB*DOC.RFFTB/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: RFFTF,RFFTI
- RFFTF** Computes Fourier coefficients of real periodic sequence (fast). Performs Fourier analysis. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): J1a1 | Usage: CALL RFFTF(N,R,WSAVE) | On-line doc: CALL GAMSDOC RFFTF (or @PRT CMLIB*DOC.RFFTF/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: RFFTB,RFFTI
- RFFTI** Initialize WSAVE array for SUBROUTINE RFFTF and RFFTB. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): -0- | Usage: CALL RFTI(N,WSAVE) | On-line doc: CALL GAMSDOC RFFTI (or @PRT CMLIB*DOC.RFFTI/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: RFFTF,RFFTB
- RG** Computes eigenvalues and, optionally, eigenvectors of a real general matrix. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a2 | Usage: CALL RG(NM,N,A,WR,WI,MATZ,Z,IV1,FV1,IERR) | On-line doc: CALL GAMSDOC RG (or @PRT CMLIB*DOC.RG/EISPACK) | Access: LIB NBS*CMLIB
- RGAUSS** Normal random number. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): L6a14 | Usage: Y = RGAUSS (XMEAN,SD) | On-line doc: CALL GAMSDOC RGAUSS (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- RG** Computes eigenvalues and eigenvectors for real generalized eigenproblem: $A \cdot X = (\text{LAMBDA}) \cdot B \cdot X$. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4b2 | Usage: CALL RGG(NM,N,A,B,ALFR,ALFI,BETA,MATZ,Z,IERR) | On-line doc: CALL GAMSDOC RGG (or @PRT CMLIB*DOC.RGG/EISPACK) | Access: LIB NBS*CMLIB
- RGM** Computes estimates of simple linear regression parameters for a geometric mean regression. | Portable single precision Fortran subprogram in SLRPACK sublibrary of CMLIB library. | Class(es): L8a1c | Usage: CALL RGM(DEBUG,IERR,N,AGM,BGM,EPS,SDA,SDB,XBAR,YBAR,X,X2,Y,Y2) | On-line doc: CALL GAMSDOC RGM (or @PRT CMLIB*DOC.RGM/SLRPACK) | Tests: CMLIB*TEST-SOURCE.RGM/SLRPACK | Access: LIB NBS*CMLIB
- RLCOMP** Generation of an orthogonal central composite design. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a6 | Usage: CALL RLCOMP (N,XMNX,IX,IOP,M,DSN,IS,IERR) | On-line doc: CALL GAMSDOC RLCOMP (or @PRT IMSL*DOC.RLCOMP) | Access: LIB NBS*IMSL
- RLDCQM** Decoding of a quadratic regression model. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8h | Usage: CALL RLDCQM (XBAR,M,BN) | On-line doc: CALL GAMSDOC RLDCQM (or @PRT IMSL*DOC.RLDCQM) | Access: LIB NBS*IMSL
- RLDCVA** Variance estimates for decoded orthogonal polynomial regression coefficients. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a2b1 L8a2b2 | Usage: CALL RLDCVA (V,ID,A,B,SC,T,IT,IERR) | On-line doc: CALL GAMSDOC RLDCVA (or @PRT IMSL*DOC.RLDCVA) | Access: LIB NBS*IMSL | See also: RLDCW
- RLDCW** Variances of coded orthogonal polynomial regression coefficients - for usage in conjunction with IMSL routines RLFOH and RLFOH, and provided to prepare input for IMSL routine RLDCVA. | Proprietary single precision Fortran subprogram in IMSL

library. | Class(es): L8h | Usage: CALL RLDCW (SSE,X,W,N,ID,IOPT,A,B,V,P,IP,IER) | On-line doc: CALL GAMSDOC RLDCW (or @PRT IMSL*DOC.RLDCW) | Access: LIB NBS*IMSL | See also: RLFOTH RLFOTW RLDCVA

RLDOPM Coefficient decoder for an orthogonal polynomial regression model. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8h | Usage: CALL RLDOPM (C,ID,A,B,T) | On-line doc: CALL GAMSDOC RLDOPM (or @PRT IMSL*DOC.RLDOPM) | Access: LIB NBS*IMSL

RLEAP Leaps and bounds algorithm for determining a number of best regression subsets from a full regression model. USLEAP is a special purpose output routine designed to be used only in conjunction with RLEAP. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a5 | Usage: CALL RLEAP (RR,KZ,IJOB,IXS,STAT,IXV,NVAR,IXB,BEST,IB,WK,IW, IER) | On-line doc: CALL GAMSDOC RLEAP (or @PRT IMSL*DOC.RLEAP) | Access: LIB NBS*IMSL

RLFITI Pure replication error degrees of freedom and sum of squares - in-core version. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a10a | Usage: CALL RLFITI (X,N,M,IX,Y,L,IY,SS,NDF,IER) | On-line doc: CALL GAMSDOC RLFITI (or @PRT IMSL*DOC.RLFITI) | Access: LIB NBS*IMSL

RLFITO Pure replication error degrees of freedom and sum of squares - out-of-core version. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a10a | Usage: CALL RLFITO (X,N,M,I,Y,L,T,W,S,ND,IER) | On-line doc: CALL GAMSDOC RLFITO (or @PRT IMSL*DOC.RLFITO) | Access: LIB NBS*IMSL

RLFOR Fit a univariate curvilinear regression model using orthogonal polynomials with optional weighting and prediction analysis - easy-to-use version. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a2b | Usage: CALL RLFOR (XYW,IX,N,RSQ,MDP,ALBP,ANOVA,B,IB,PRED,IP,WK,IER) | On-line doc: CALL GAMSDOC RLFOR (or @PRT IMSL*DOC.RLFOR) | Access: LIB NBS*IMSL | See also: BDTRGI

RLFOTH Fit a univariate curvilinear regression model using orthogonal polynomials. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a2b1 | Usage: CALL RLFOTH (X,Y,N,RSQ,MD,ID,P,C,S,A,B,IER) | On-line doc: CALL GAMSDOC RLFOTH (or @PRT IMSL*DOC.RLFOTH) | Access: LIB NBS*IMSL | See also: RLDOPM

RLFOTW Fit a univariate curvilinear regression model using orthogonal polynomials with weighting. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a2b2 | Usage: CALL RLFOTW (X,Y,N,RSQ,MD,W,ID,P,C,S,A,B,IER) | On-line doc: CALL GAMSDOC RLFOTW (or @PRT IMSL*DOC.RLFOTW) | Access: LIB NBS*IMSL | See also: RLDOPM

RLGQMI Centering of independent variable settings and generation of centered square and cross product terms - in-core version. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2a | Usage: CALL RLGQMI (X,N,M,IX,XBAR) | On-line doc: CALL GAMSDOC RLGQMI (or @PRT IMSL*DOC.RLGQMI) | Access: LIB NBS*IMSL

RLGQMO Centering of independent variable settings and generation of uncentered square and cross product terms - out-of-core version. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2a | Usage: CALL RLGQMO (X,N,M,I,XBAR,IER) | On-line doc: CALL GAMSDOC RLGQMO (or @PRT IMSL*DOC.RLGQMO) | Access: LIB NBS*IMSL

RLINCF Response control using a fitted simple linear regression model. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a1d | Usage: CALL RLINCF (CRIT,IOP,STAT,IER) | On-line doc: CALL GAMSDOC RLINCF (or @PRT IMSL*DOC.RLINCF) | Access: LIB NBS*IMSL

RLINE Fits straight line to x-y data by resistant line procedure - partitions data by x-value into three groups and uses an iterative procedure to find the line that makes the median residual in the left and the right partitions equal. | Command in MINITAB Proprietary interactive system. Class(es): L8f | Usage: RLINE y in C, x in C [put residuals into C [predicted values in C [coefficients into C]]] ; subcommand MAXITERATION = K | On-line doc: HELP RLINE (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

RLINPF Inverse prediction using a fitted simple linear regression model. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a10c | Usage: CALL RLINPF (CRIT,IOP,STAT,IER) | On-line doc: CALL GAMSDOC RLINPF (or @PRT IMSL*DOC.RLINPF) | Access: LIB NBS*IMSL

RLLAV Perform linear regression using the least absolute values criterion. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8d K3 | Usage: CALL RLLAV (XY,IXY,N,M,IOPT,BETA,SUMRE,ITER,IRANK,IWK,WK,IER) | On-line doc: CALL GAMSDOC RLLAV (or @PRT IMSL*DOC.RLLAV) | Access: LIB NBS*IMSL

RLLMV Perform linear regression using the minimax criterion. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8c K2 | Usage: CALL RLLMV (XY,IXY,N,M,IOPT,BETA,REMAX,ITER,IRANK,WK,IER) | On-line doc: CALL GAMSDOC RLLMV (or @PRT IMSL*DOC.RLLMV) | Access: LIB NBS*IMSL

RLMUL Multiple linear regression analysis. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a4a1c | Usage: CALL RLMUL (A,XYBAR,N,M,ALFA,ANOVA,B,IB,VARB,IER) | On-line doc: CALL GAMSDOC RLMUL (or @PRT IMSL*DOC.RLMUL) | Access: LIB NBS*IMSL | See also: BECOVM RLSUM

RLONE Analysis of a simple linear regression model. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a1a1a | Usage: CALL RLONE (XY,IX,N,IMOD,IPRED,ALBAP,DES,ANOVA,STAT,PRED,IP,NN, IER) | On-line doc: CALL GAMSDOC RLONE (or @PRT IMSL*DOC.RLONE) | Access: LIB NBS*IMSL

- RLOPDC** Response prediction using an orthogonal polynomial regression model. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a10c | Usage: CALL RLOPDC (X,N,A,B,C,ID,IOPT,P,YHAT,IER) | On-line doc: CALL GAMSDOC RLOPDC (or @PRT IMSL*DOC.RLOPDC) | Access: LIB NBS*IMSL
- RLPOL** Generate orthogonal polynomials with the associated constants AA and BB. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a2b L2a C3a2 | Usage: CALL RLPOL (X,N,ID,SM,SA,AA,BB,P,IP,IER) | On-line doc: CALL GAMSDOC RLPOL (or @PRT IMSL*DOC.RLPOL) | Access: LIB NBS*IMSL
- RLPRDI** Confidence intervals for the true response and for the average of a set of future observations on the response - in-core version. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a10c | Usage: CALL RLPRDI (Y,V,N,TS,NR,C,IC) | On-line doc: CALL GAMSDOC RLPRDI (or @PRT IMSL*DOC.RLPRDI) | Access: LIB NBS*IMSL
- RLPRDO** Confidence intervals for the true response and for the average of a set of future observations on the response - out-of-core version. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a10c | Usage: CALL RLPRDO (Y,V,N,TS,NR,C) | On-line doc: CALL GAMSDOC RLPRDO (or @PRT IMSL*DOC.RLPRDO) | Access: LIB NBS*IMSL
- RLRES** Perform a residual analysis for a fitted regression model. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a10b | Usage: CALL RLRES (XY,IX,MM,N,IH,M,BETA,SDR,RES,IR,IER) | On-line doc: CALL GAMSDOC RLRES (or @PRT IMSL*DOC.RLRES) | Access: LIB NBS*IMSL
- RLSEP** Selection of a regression model using a forward stepwise algorithm, and computation of the usual analysis of variance table entries - easy-to-use version. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a5 | Usage: CALL RLSEP (XY,N,M,IX,ALFA,IJOB,IND,ANOVA,XYB,IB,VARB,IER) | On-line doc: CALL GAMSDOC RLSEP (or @PRT IMSL*DOC.RLSEP) | Access: LIB NBS*IMSL
- RLSTP** Regression model selection using a forward stepwise algorithm with results available after each step. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a5 | Usage: CALL RLSTP (A,M,N,ALFA,ALFAO,JX,IH,B,IOPT,IER) | On-line doc: CALL GAMSDOC RLSTP (or @PRT IMSL*DOC.RLSTP) | Access: LIB NBS*IMSL
- RLSUBM** Retrieval of a symmetric submatrix from a matrix stored in symmetric storage mode by RLSTP. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8h | Usage: CALL RLSUBM (A,M,IH,S,N) | On-line doc: CALL GAMSDOC RLSUBM (or @PRT IMSL*DOC.RLSUBM) | Access: LIB NBS*IMSL | See also: RLSTP
- RLSUM** Reordering of the rows and corresponding columns of a symmetric matrix stored in symmetric storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8h | Usage: CALL RLSUM (AA,MM,IH,M,A,IER) | On-line doc: CALL GAMSDOC RLSUM (or @PRT IMSL*DOC.RLSUM) | Access: LIB NBS*IMSL
- RLTR** An auxiliary routine for use together with FFT to transform 2N real data points. Uses less storage than FFTR. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DRLTR. | Class(es): J1a1 | Usage: CALL RLTR (A,B,N,ISN) | On-line doc: CALL GAMSDOC RLTR (or @PRT PORT*DOC.RLTR) | Access: LIB NBS*PORT | See also: FFT
- RNOR** Generates quasi normal random numbers with zero mean and unit standard deviation. | Portable single precision Fortran subprogram in CMLIB library. | Class(es): L6a14 | Usage: S=RNOR(JD) | On-line doc: CALL GAMSDOC RNOR (or @PRT CMLIB*DOC.RNOR/RV) | Tests: CMLIB*TEST-SOURCE.RNOR/RV | Access: LIB NBS*CMLIB
- ROOTOGRAM** Prints a suspended rootogram, i.e. a histogram which has been fit with a Gaussian distribution based on square roots of the counts of data values in each bin and which uses medians and hinges. Options: specify Gaussian mean and standard deviation, save results. | Command in MINITAB Proprietary interactive system. | Class(es): L3d | Usage: ROOTogram data in C [use bin boundaries in C] ; subcommands BOUNDARIES into C; DRRS into C; FITTED values into C; COUNTS into C; FREQUENCIES are in C [bin boundaries are in C]; MEAN = K; STDEV = K. | On-line doc: HELP ROOTOGRAM (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- ROTN** Plane rotation of two rows of a matrix. | Portable single precision Fortran subprogram in NASHLIB sublibrary of MATHWARE library. | Class(es): D1b10 | Usage: CALL ROTN(J,K,S,C,M,T,W,ND1,NG) | On-line doc: @PRT,S MATHWARE*NASHLIB.A4A/ROTN | Tests: MATHWARE*NASHLIB.A4-4A | Access: See individual sublibrary documentation
- RPOLY** Finds zeros of a polynomial with real coefficients. Output zeros are in a pair of arrays, for real and imaginary part. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): F1a1a | Usage: CALL RPOLY (DEGREE,COEFF,ZEROR,ZEROI) | On-line doc: CALL GAMSDOC RPOLY (or @PRT PORT*DOC.RPOLY) | Access: LIB NBS*PORT
- RPQR79** Computes all the zeros of a general real polynomial using eigenvalue methods, requiring NxN storage for Nth degree polynomial. | Portable single precision Fortran subprogram in CPQR79 sublibrary of CMLIB library. | Class(es): F1a1a | Usage: CALL RPQR79(NDEG,COEFF,ROOT,IERR,WORK) | On-line doc: CALL GAMSDOC RPQR79 (or @PRT CMLIB*DOC.RPQR79/CPQR79 and CMLIB*DOC.SUMMARY/CPQR79) | Tests: CMLIB*TEST-SOURCE.\$F/CPQR79 | Access: LIB NBS*CMLIB
- RPZERO** Computes all the zeros of a polynomial with real coefficients. Error bounds are also computed. Uses Newton's Method for systems. | Portable single precision Fortran subprogram in CPZERO sublibrary of CMLIB library. | Class(es): F1a1a | Usage: CALL RPZERO(N,A,R,T,IFLAG,S) | On-line doc: CALL GAMSDOC RPZERO (or @PRT CMLIB*DOC.SUMMARY/CPZERO and CMLIB*DOC.RPZERO/CPZERO) | Tests: CMLIB*TEST-SOURCE.\$Q/CPZERO | Access: LIB NBS*CMLIB
- RQUAD** Finds the integral of a general user defined EXTERNAL function by an adaptive technique. Combined absolute and rela-

- tive error control. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DRQUAD. | Class(es): H2a1a1 | Usage: CALL RQUAD (F,A,B,EPSABS,EPSREL,ANS,ERREST) | On-line doc: CALL GAMSDOC RQUAD (or @PRT PORT*DOC.RQUAD) | Access: LIB NBS*PORT
- RS** Computes eigenvalues and, optionally, eigenvectors of a real symmetric matrix. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a1 | Usage: CALL RS(NM,N,A,W,MATZ,Z,FV1,FV2,IERR) | On-line doc: CALL GAMSDOC RS (or @PRT CMLIB*DOC.RS/EISPACK) | Access: LIB NBS*CMLIB
- RSB** Computes eigenvalues and, optionally, eigenvectors of real symmetric band matrix. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a6 | Usage: CALL RSB(NM,N,MB,A,W,MATZ,Z,FV1,FV2,IERR) | On-line doc: CALL GAMSDOC RSB (or @PRT CMLIB*DOC.RSB/EISPACK) | Access: LIB NBS*CMLIB
- RSCG** Iterative solution of large sparse systems of linear equations. Reduced system method, conjugate gradient acceleration, adaptive. | Portable single precision Fortran subprogram in ITPACK sublibrary of MATHWARE library. | Class(es): D2b4 I2b4b D2a4 | Usage: CALL RSCG(N,IA,JA,A,RHS,U,IWKSP,NW,WKSP,IPARM,RPARM,IER) | On-line doc: @PRT,S MATHWARE*ITPACK.DOCUMENT | Access: See individual sublibrary documentation
- RSG** Computes eigenvalues and, optionally, eigenvectors of real symmetric generalized eigenproblem: $A*X=(LAMBDA)*B*X$. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4b1 | Usage: CALL RSG(NM,N,A,B,W,MATZ,Z,FV1,FV2,IERR) | On-line doc: CALL GAMSDOC RSG (or @PRT CMLIB*DOC.RSG/EISPACK) | Access: LIB NBS*CMLIB
- RSGAB** Computes eigenvalues and, optionally, eigenvectors of real symmetric generalized eigenproblem: $A*B*X=(LAMBDA)*X$. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4b1 | Usage: CALL RSGAB(NM,N,A,B,W,MATZ,Z,FV1,FV2,IERR) | On-line doc: CALL GAMSDOC RSGAB (or @PRT CMLIB*DOC.RSGAB/EISPACK) | Access: LIB NBS*CMLIB
- RSGBA** Computes eigenvalues and, optionally, eigenvectors of real symmetric generalized eigenproblem: $B*A*X=(LAMBDA)*X$. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4b1 | Usage: CALL RSGBA(NM,N,A,B,W,MATZ,Z,FV1,FV2,IERR) | On-line doc: CALL GAMSDOC RSGBA (or @PRT CMLIB*DOC.RSGBA/EISPACK) | Access: LIB NBS*CMLIB
- RSMITZ** Least squares fit of the non-linear regression model $y(i) = \alpha + \beta * \gamma^{*x(i)} + e(i)$. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8g1a | Usage: CALL RSMITZ (XY,IXY,IV,PAR,STAT,VCV,ITER,IER) | On-line doc: CALL GAMSDOC RSMITZ (or @PRT IMSL*DOC.RSMITZ) | Access: LIB NBS*IMSL
- RSMOOTH** Computes resistant smoother by 4253H, twice (or 3RSSH, twice), i.e. successive application of running medians and Hanning (running weighted averages), and save results. | Command in MINITAB Proprietary interactive system. Class(es): L8f | Usage: RSMOoth C, put rough into C, smooth into C [; subcommand SMOOTH 3RSSH, twice.] | On-line doc: HELP RSMOOTH (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- RSP** Compute eigenvalues and, optionally, eigenvectors of a real symmetric matrix packed into a one dimensional array. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a1 | Usage: CALL RSP(NM,N,NV,A,W,MATZ,Z,FV1,FV2,IERR) | On-line doc: CALL GAMSDOC RSP (or @PRT CMLIB*DOC.RSP/EISPACK) | Access: LIB NBS*CMLIB
- RSSI** Iterative solution of large sparse systems of linear equations. Reduced system method, Chebyshev acceleration, adaptive. | Portable single precision Fortran subprogram in ITPACK sublibrary of MATHWARE library. | Class(es): D2b4 I2b4b D2a4 | Usage: CALL RSSI(N,IA,JA,A,RHS,U,IWKSP,NW,WKSP,IPARM,RPARM,IER) | On-line doc: @PRT,S MATHWARE*ITPACK.DOCUMENT | Access: See individual sublibrary documentation
- RST** Compute eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 | Usage: CALL RST(NM,N,W,E,MATZ,Z,IERR) | On-line doc: CALL GAMSDOC RST (or @PRT CMLIB*DOC.RST/EISPACK) | Access: LIB NBS*CMLIB
- RT** Compute eigenvalues and eigenvectors of a special real tridiagonal matrix. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 | Usage: CALL RT(NM,N,A,W,MATZ,Z,FV1,IERR) | On-line doc: CALL GAMSDOC RT (or @PRT CMLIB*DOC.RT/EISPACK) | Access: LIB NBS*CMLIB
- RUNIF** Sequence of uniform random numbers on [0,1]. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): L6a21 | Usage: Y = RUNIF (T,N) | On-line doc: CALL GAMSDOC RUNIF (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- RUNS** Performs a run analysis of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L4a1d | Usage: CALL RUNS(X,N) | On-line doc: CALL GAMSDOC RUNS (or @PRT DATAPAC*DOC.RUNS) | Access: LIB NBS*DAPAC
- RUNS** Performs a two-sided runs test. | Command in MINITAB Proprietary interactive system. Class(es): L4a1d | Usage: RUNS above and below K for data in column C | On-line doc: HELP RUNS (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT

NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

RYORK Estimates simple linear regression coefficients when both variables are subject to errors which are not necessarily homogeneous in variance. | Portable single precision Fortran subprogram in SLRPACK sublibrary of CMLIB library. | Class(es): L8a1c | Usage: CALL RYORK(DEBUG,IER,IRES,ITER,MXITER,N,BSTART,DELMAX,EPS,SDA, SDB,A, ANGLE,B,U, U2,UV,V,V2,W,W3,WWX,WWY,WX,WY,X,XBAR,XRES,Y,YRYBAR, YRES) | On-line doc: CALL GAMSDOC RYORK (or @PRT CMLIB*DOC.RYORK/SLRPACK) | Tests: CMLIB*TEST-SOURCE.RYORK/SLRPACK | Access: LIB NBS*CMLIB | See also: RGM

S

- S07AAE** $\text{Tan}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S07AAF. | Class(es): C4a | Usage: D = S07AAE (X, IFAIL) | On-line doc: CALL GAMSDOC S07AAE (or @PRT NAG*DOC.S07AAE) | Access: LIB NBS*NAG
- S07AAF** $\text{Tan}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S07AAE. | Class(es): C4a | Usage: D = S07AAF (X, IFAIL) | On-line doc: CALL GAMSDOC S07AAF (or @PRT NAG*DOC.S07AAF) | Access: LIB NBS*NAG
- S09AAE** $\text{Arcsin}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S09AAF. | Class(es): C4a | Usage: D = S09AAE (X, IFAIL) | On-line doc: CALL GAMSDOC S09AAE (or @PRT NAG*DOC.S09AAE) | Access: LIB NBS*NAG
- S09AAF** $\text{Arcsin}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S09AAE. | Class(es): C4a | Usage: D = S09AAF (X, IFAIL) | On-line doc: CALL GAMSDOC S09AAF (or @PRT NAG*DOC.S09AAF) | Access: LIB NBS*NAG
- S09ABE** $\text{Arccos}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S09ABF. | Class(es): C4a | Usage: D = S09ABE (X, IFAIL) | On-line doc: CALL GAMSDOC S09ABE (or @PRT NAG*DOC.S09ABE) | Access: LIB NBS*NAG
- S09ABF** $\text{Arccos}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S09ABE. | Class(es): C4a | Usage: D = S09ABF (X, IFAIL) | On-line doc: CALL GAMSDOC S09ABF (or @PRT NAG*DOC.S09ABF) | Access: LIB NBS*NAG
- S10AAE** $\text{Tanh}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S10AAF. | Class(es): C4c | Usage: D = S10AAE (X, IFAIL) | On-line doc: CALL GAMSDOC S10AAE (or @PRT NAG*DOC.S10AAE) | Access: LIB NBS*NAG
- S10AAF** $\text{Tanh}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S10AAE. | Class(es): C4c | Usage: D = S10AAF (X, IFAIL) | On-line doc: CALL GAMSDOC S10AAF (or @PRT NAG*DOC.S10AAF) | Access: LIB NBS*NAG
- S10ABE** $\text{Sinh}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S10ABF. | Class(es): C4c | Usage: D = S10ABE (X, IFAIL) | On-line doc: CALL GAMSDOC S10ABE (or @PRT NAG*DOC.S10ABE) | Access: LIB NBS*NAG
- S10ABF** $\text{Sinh}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S10ABE. | Class(es): C4c | Usage: D = S10ABF (X, IFAIL) | On-line doc: CALL GAMSDOC S10ABF (or @PRT NAG*DOC.S10ABF) | Access: LIB NBS*NAG
- S10ACE** $\text{Cosh}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S10ACF. | Class(es): C4c | Usage: D = S10ACE (X, IFAIL) | On-line doc: CALL GAMSDOC S10ACE (or @PRT NAG*DOC.S10ACE) | Access: LIB NBS*NAG
- S10ACF** $\text{Cosh}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S10ACE. | Class(es): C4c | Usage: D = S10ACF (X, IFAIL) | On-line doc: CALL GAMSDOC S10ACF (or @PRT NAG*DOC.S10ACF) | Access: LIB NBS*NAG
- S11AAE** $\text{Arctanh}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S11AAF. | Class(es): C4c | Usage: D = S11AAE (X, IFAIL) | On-line doc: CALL GAMSDOC S11AAE (or @PRT NAG*DOC.S11AAE) | Access: LIB NBS*NAG
- S11AAF** $\text{Arctanh}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S11AAE. | Class(es): C4c | Usage: D = S11AAF (X, IFAIL) | On-line doc: CALL GAMSDOC S11AAF (or @PRT NAG*DOC.S11AAF) | Access: LIB NBS*NAG
- S11ABE** $\text{Arcsinh}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S11ABF. | Class(es): C4c | Usage: D = S11ABE (X, IFAIL) | On-line doc: CALL GAMSDOC S11ABE (or @PRT NAG*DOC.S11ABE) | Access: LIB NBS*NAG
- S11ABF** $\text{Arcsinh}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S11ABE. | Class(es): C4c | Usage: D = S11ABF (X, IFAIL) | On-line doc: CALL GAMSDOC S11ABF (or @PRT NAG*DOC.S11ABF) | Access: LIB NBS*NAG
- S11ACE** $\text{Arccosh}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S11ACF. | Class(es): C4c | Usage: D = S11ACE (X, IFAIL) | On-line doc: CALL GAMSDOC S11ACE (or @PRT NAG*DOC.S11ACE) | Access: LIB NBS*NAG
- S11ACF** $\text{Arccosh}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S11ACE. | Class(es): C4c | Usage: D = S11ACF (X, IFAIL) | On-line doc: CALL GAMSDOC S11ACF (or @PRT NAG*DOC.S11ACF) | Access: LIB NBS*NAG
- S13AAE** Exponential integral, $\text{ei}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S13AAF. | Class(es): C5 | Usage: D = S13AAE (X, IFAIL) | On-line doc: CALL GAMSDOC S13AAE (or @PRT NAG*DOC.S13AAE) | Access: LIB NBS*NAG
- S13AAF** Exponential integral, $\text{ei}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S13AAE. | Class(es): C5 | Usage: D = S13AAF (X, IFAIL) | On-line doc: CALL GAMSDOC S13AAF (or @PRT NAG*DOC.S13AAF) | Access: LIB NBS*NAG
- S13ACE** Cosine integral, $\text{ci}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S13ACF. | Class(es): C6 | Usage: D = S13ACE (X, IFAIL) | On-line doc: CALL GAMSDOC S13ACE (or @PRT NAG*DOC.S13ACE) | Access: LIB NBS*NAG
- S13ACF** Cosine integral, $\text{ci}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S13ACE. | Class(es): C6 | Usage: D = S13ACF (X, IFAIL) | On-line doc: CALL GAMSDOC S13ACF (or @PRT NAG*DOC.S13ACF) | Access: LIB NBS*NAG
- S13ADE** Sine integral, $\text{si}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S13ADF. | Class(es): C6 | Usage: D = S13ADE (X, IFAIL) | On-line doc: CALL GAMSDOC S13ADE (or @PRT NAG*DOC.S13ADE) | Access:

LIB NBS*NAG

- S13ADF** Sine integral, $\text{si}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S13ADE. | Class(es): C6 | Usage: D = S13ADF (X, IFAIL) | On-line doc: CALL GAMSDOC S13ADF (or @PRT NAG*DOC.S13ADF) | Access: LIB NBS*NAG
- S14AAE** Gamma function. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S14AAF. | Class(es): C7a | Usage: D = S14AAE (X, IFAIL) | On-line doc: CALL GAMSDOC S14AAE (or @PRT NAG*DOC.S14AAE) | Access: LIB NBS*NAG
- S14AAF** Gamma function. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S14AAE. | Class(es): C7a | Usage: D = S14AAF (X, IFAIL) | On-line doc: CALL GAMSDOC S14AAF (or @PRT NAG*DOC.S14AAF) | Access: LIB NBS*NAG
- S14ABE** Log gamma function. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S14ABF. | Class(es): C7a | Usage: D = S14ABE (X, IFAIL) | On-line doc: CALL GAMSDOC S14ABE (or @PRT NAG*DOC.S14ABE) | Access: LIB NBS*NAG
- S14ABF** Log gamma function. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S14ABE. | Class(es): C7a | Usage: D = S14ABF (X, IFAIL) | On-line doc: CALL GAMSDOC S14ABF (or @PRT NAG*DOC.S14ABF) | Access: LIB NBS*NAG
- S15ABE** Cumulative normal distribution function, $p(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S15ABF. | Class(es): C8a L5a1n | Usage: D = S15ABE (X, IFAIL) | On-line doc: CALL GAMSDOC S15ABE (or @PRT NAG*DOC.S15ABE) | Access: LIB NBS*NAG
- S15ABF** Cumulative normal distribution function, $p(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S15ABE. | Class(es): C8a L5a1n | Usage: D = S15ABF (X, IFAIL) | On-line doc: CALL GAMSDOC S15ABF (or @PRT NAG*DOC.S15ABF) | Access: LIB NBS*NAG
- S15ACE** Complement of cumulative normal distribution function, $q(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S15ACF. | Class(es): C8a L5a1n | Usage: D = S15ACE (X, IFAIL) | On-line doc: CALL GAMSDOC S15ACE (or @PRT NAG*DOC.S15ACE) | Access: LIB NBS*NAG
- S15ACF** Complement of cumulative normal distribution function, $q(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S15ACE. | Class(es): C8a L5a1n | Usage: D = S15ACF (X, IFAIL) | On-line doc: CALL GAMSDOC S15ACF (or @PRT NAG*DOC.S15ACF) | Access: LIB NBS*NAG
- S15ADE** Complement of error function, $\text{erfc}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S15ADF. | Class(es): C8a L5a1e | Usage: D = S15ADE (X, IFAIL) | On-line doc: CALL GAMSDOC S15ADE (or @PRT NAG*DOC.S15ADE) | Access: LIB NBS*NAG
- S15ADF** Complement of error function, $\text{erfc}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S15ADE. | Class(es): C8a L5a1e | Usage: D = S15ADF (X, IFAIL) | On-line doc: CALL GAMSDOC S15ADF (or @PRT NAG*DOC.S15ADF) | Access: LIB NBS*NAG
- S15AEE** Error function, $\text{erf}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S15AEF. | Class(es): C8a L5a1e | Usage: D = S15AEE (X, IFAIL) | On-line doc: CALL GAMSDOC S15AEE (or @PRT NAG*DOC.S15AEE) | Access: LIB NBS*NAG | See also: S15ADE
- S15AEF** Error function, $\text{erf}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S15AEE. | Class(es): C8a L5a1e | Usage: D = S15AEF (X, IFAIL) | On-line doc: CALL GAMSDOC S15AEF (or @PRT NAG*DOC.S15AEF) | Access: LIB NBS*NAG | See also: S15ADF
- S15AFE** Dawson's integral. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S15AFF. | Class(es): C8c | Usage: D = S15AFE (X, IFAIL) | On-line doc: CALL GAMSDOC S15AFE (or @PRT NAG*DOC.S15AFE) | Access: LIB NBS*NAG
- S15AFF** Dawson's integral. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S15AFE. | Class(es): C8c | Usage: D = S15AFF (X, IFAIL) | On-line doc: CALL GAMSDOC S15AFF (or @PRT NAG*DOC.S15AFF) | Access: LIB NBS*NAG
- S17ACE** Bessel functions, $y_0(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S17ACF. | Class(es): C10a1 | Usage: D = S17ACE (X, IFAIL) | On-line doc: CALL GAMSDOC S17ACE (or @PRT NAG*DOC.S17ACE) | Access: LIB NBS*NAG
- S17ACF** Bessel functions, $y_0(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S17ACE. | Class(es): C10a1 | Usage: D = S17ACF (X, IFAIL) | On-line doc: CALL GAMSDOC S17ACF (or @PRT NAG*DOC.S17ACF) | Access: LIB NBS*NAG
- S17ADE** Bessel functions, $y_1(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S17ADF. | Class(es): C10a1 | Usage: D = S17ADE (X, IFAIL) | On-line doc: CALL GAMSDOC S17ADE (or @PRT NAG*DOC.S17ADE) | Access: LIB NBS*NAG
- S17ADF** Bessel functions, $y_1(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S17ADE. | Class(es): C10a1 | Usage: D = S17ADF (X, IFAIL) | On-line doc: CALL GAMSDOC S17ADF (or @PRT NAG*DOC.S17ADF) | Access: LIB NBS*NAG

- S17AEE** Bessel functions, $j_0(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S17AEF. |
 Class(es): C10a1 | Usage: D = S17AEE (X, IFAIL) | On-line doc: CALL GAMSDOC S17AEE (or @PRT NAG*DOC.S17AEE) | Access:
 LIB NBS*NAG
- S17AEF** Bessel functions, $j_0(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S17AEE. |
 Class(es): C10a1 | Usage: D = S17AEF (X, IFAIL) | On-line doc: CALL GAMSDOC S17AEF (or @PRT NAG*DOC.S17AEF) | Access:
 LIB NBS*NAG
- S17AFE** Bessel functions, $j_1(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S17AFF. |
 Class(es): C10a1 | Usage: D = S17AFE (X, IFAIL) | On-line doc: CALL GAMSDOC S17AFE (or @PRT NAG*DOC.S17AFE) | Access:
 LIB NBS*NAG
- S17AFF** Bessel functions, $j_1(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S17AFE. |
 Class(es): C10a1 | Usage: D = S17AFF (X, IFAIL) | On-line doc: CALL GAMSDOC S17AFF (or @PRT NAG*DOC.S17AFF) | Access:
 LIB NBS*NAG
- S17AGE** Airy functions, $ai(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S17AGF. |
 Class(es): C10d | Usage: D = S17AGE (X, IFAIL) | On-line doc: CALL GAMSDOC S17AGE (or @PRT NAG*DOC.S17AGE) | Access:
 LIB NBS*NAG
- S17AGF** Airy functions, $ai(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S17AGE. |
 Class(es): C10d | Usage: D = S17AGF (X, IFAIL) | On-line doc: CALL GAMSDOC S17AGF (or @PRT NAG*DOC.S17AGF) | Access:
 LIB NBS*NAG
- S17AHE** Airy functions, $bi(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S17AHF. |
 Class(es): C10d | Usage: D = S17AHE (X, IFAIL) | On-line doc: CALL GAMSDOC S17AHE (or @PRT NAG*DOC.S17AHE) | Access:
 LIB NBS*NAG
- S17AHF** Airy functions, $bi(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S17AHE. |
 Class(es): C10d | Usage: D = S17AHF (X, IFAIL) | On-line doc: CALL GAMSDOC S17AHF (or @PRT NAG*DOC.S17AHF) | Access:
 LIB NBS*NAG
- S17AJE** Airy functions, $ai'(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S17AJF. |
 Class(es): C10d | Usage: D = S17AJE (X, IFAIL) | On-line doc: CALL GAMSDOC S17AJE (or @PRT NAG*DOC.S17AJE) | Access:
 LIB NBS*NAG
- S17AJF** Airy functions, $ai'(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S17AJE. |
 Class(es): C10d | Usage: D = S17AJF (X, IFAIL) | On-line doc: CALL GAMSDOC S17AJF (or @PRT NAG*DOC.S17AJF) | Access:
 LIB NBS*NAG
- S17AKE** Airy functions, $bi'(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S17AKF. |
 Class(es): C10d | Usage: D = S17AKE (X, IFAIL) | On-line doc: CALL GAMSDOC S17AKE (or @PRT NAG*DOC.S17AKE) | Access:
 LIB NBS*NAG
- S17AKF** Airy functions, $bi'(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S17AKE. |
 Class(es): C10d | Usage: D = S17AKF (X, IFAIL) | On-line doc: CALL GAMSDOC S17AKF (or @PRT NAG*DOC.S17AKF) | Access:
 LIB NBS*NAG
- S18ACE** Modified Bessel functions, $K_0(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is
 S18ACF. | Class(es): C10b1 | Usage: D = S18ACE (X, IFAIL) | On-line doc: CALL GAMSDOC S18ACE (or @PRT NAG*DOC.S18ACE)
 | Access: LIB NBS*NAG
- S18ACF** Modified Bessel functions, $K_0(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is
 S18ACE. | Class(es): C10b1 | Usage: D = S18ACF (X, IFAIL) | On-line doc: CALL GAMSDOC S18ACF (or @PRT NAG*DOC.S18ACF)
 | Access: LIB NBS*NAG
- S18ADE** Modified Bessel functions, $K_1(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is
 S18ADF. | Class(es): C10b1 | Usage: D = S18ADE (X, IFAIL) | On-line doc: CALL GAMSDOC S18ADE (or @PRT NAG*DOC.S18ADE)
 | Access: LIB NBS*NAG
- S18ADF** Modified Bessel functions, $K_1(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is
 S18ADE. | Class(es): C10b1 | Usage: D = S18ADF (X, IFAIL) | On-line doc: CALL GAMSDOC S18ADF (or @PRT NAG*DOC.S18ADF)
 | Access: LIB NBS*NAG
- S18AEE** Modified Bessel functions, $I_0(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is
 S18AEF. | Class(es): C10b1 | Usage: D = S18AEE (X, IFAIL) | On-line doc: CALL GAMSDOC S18AEE (or @PRT NAG*DOC.S18AEE)
 | Access: LIB NBS*NAG
- S18AEF** Modified Bessel functions, $I_0(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is
 S18AEE. | Class(es): C10b1 | Usage: D = S18AEF (X, IFAIL) | On-line doc: CALL GAMSDOC S18AEF (or @PRT NAG*DOC.S18AEF)
 | Access: LIB NBS*NAG
- S18AFE** Modified Bessel functions, $I_1(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is
 S18AFF. | Class(es): C10b1 | Usage: D = S18AFE (X, IFAIL) | On-line doc: CALL GAMSDOC S18AFE (or @PRT NAG*DOC.S18AFE)
 | Access: LIB NBS*NAG

- S18AFF** Modified Bessel functions, $I_1(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S18AFE. | Class(es): C10b1 | Usage: $D = S18AFF(X, IFAIL)$ | On-line doc: CALL GAMSDOC S18AFF (or @PRT NAG*DOC.S18AFF) | Access: LIB NBS*NAG
- S18CCE** Returns a value of the scaled modified Bessel function, $\exp(x)K_{\text{sub0}}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S18CCF. | Class(es): C10b1 | Usage: $R = S18CCE(X, IFAIL)$ | On-line doc: CALL GAMSDOC S18CCE (or @PRT NAG*DOC.S18CCE) | Access: LIB NBS*NAG
- S18CCF** Returns a value of the scaled modified Bessel function, $\exp(x)K_{\text{sub0}}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S18CCE. | Class(es): C10b1 | Usage: $R = S18CCF(X, IFAIL)$ | On-line doc: CALL GAMSDOC S18CCF (or @PRT NAG*DOC.S18CCF) | Access: LIB NBS*NAG
- S18CDE** Returns a value of the scaled modified Bessel function, $\exp(x)K_{\text{sub1}}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S18CDF. | Class(es): C10b1 | Usage: $R = S18CDE(X, IFAIL)$ | On-line doc: CALL GAMSDOC S18CDE (or @PRT NAG*DOC.S18CDE) | Access: LIB NBS*NAG
- S18CDF** Returns a value of the scaled modified Bessel function, $\exp(x)K_{\text{sub1}}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S18CDE. | Class(es): C10b1 | Usage: $R = S18CDF(X, IFAIL)$ | On-line doc: CALL GAMSDOC S18CDF (or @PRT NAG*DOC.S18CDF) | Access: LIB NBS*NAG
- S18CEE** Returns a value of the scaled modified Bessel function, $\exp(-|x|)I_{\text{sub0}}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S18CEF. | Class(es): C10b1 | Usage: $R = S18CEE(X, IFAIL)$ | On-line doc: CALL GAMSDOC S18CEE (or @PRT NAG*DOC.S18CEE) | Access: LIB NBS*NAG
- S18CEF** Returns a value of the scaled modified Bessel function, $\exp(-|x|)I_{\text{sub0}}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S18CEE. | Class(es): C10b1 | Usage: $R = S18CEF(X, IFAIL)$ | On-line doc: CALL GAMSDOC S18CEF (or @PRT NAG*DOC.S18CEF) | Access: LIB NBS*NAG
- S18CFE** Returns a value of the scaled modified Bessel function, $\exp(-|x|)I_{\text{sub1}}(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S18CFF. | Class(es): C10b1 | Usage: $R = S18CFE(X, IFAIL)$ | On-line doc: CALL GAMSDOC S18CFE (or @PRT NAG*DOC.S18CFE) | Access: LIB NBS*NAG
- S18CFF** Returns a value of the scaled modified Bessel function, $\exp(-|x|)I_{\text{sub1}}(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S18CFE. | Class(es): C10b1 | Usage: $R = S18CFF(X, IFAIL)$ | On-line doc: CALL GAMSDOC S18CFF (or @PRT NAG*DOC.S18CFF) | Access: LIB NBS*NAG
- S20ACE** Fresnel integrals, $s(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S20ACF. | Class(es): C8b | Usage: $D = S20ACE(X, IFAIL)$ | On-line doc: CALL GAMSDOC S20ACE (or @PRT NAG*DOC.S20ACE) | Access: LIB NBS*NAG
- S20ACF** Fresnel integrals, $s(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S20ACE. | Class(es): C8b | Usage: $D = S20ACF(X, IFAIL)$ | On-line doc: CALL GAMSDOC S20ACF (or @PRT NAG*DOC.S20ACF) | Access: LIB NBS*NAG
- S20ADE** Fresnel integrals, $c(x)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S20ADF. | Class(es): C8b | Usage: $D = S20ADE(X, IFAIL)$ | On-line doc: CALL GAMSDOC S20ADE (or @PRT NAG*DOC.S20ADE) | Access: LIB NBS*NAG
- S20ADF** Fresnel integrals, $c(x)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S20ADE. | Class(es): C8b | Usage: $D = S20ADF(X, IFAIL)$ | On-line doc: CALL GAMSDOC S20ADF (or @PRT NAG*DOC.S20ADF) | Access: LIB NBS*NAG
- S21BAE** Elliptic integrals, degenerate symmetrised integral of 1st kind, $rc(x, y)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S21BAF. | Class(es): C14 | Usage: $D = S21BAE(X, Y, IFAIL)$ | On-line doc: CALL GAMSDOC S21BAE (or @PRT NAG*DOC.S21BAE) | Access: LIB NBS*NAG
- S21BAF** Elliptic integrals, degenerate symmetrised integral of 1st kind, $rc(x, y)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S21BAE. | Class(es): C14 | Usage: $D = S21BAF(X, Y, IFAIL)$ | On-line doc: CALL GAMSDOC S21BAF (or @PRT NAG*DOC.S21BAF) | Access: LIB NBS*NAG
- S21BBE** Elliptic integrals, symmetrised integral of 1st kind, $rf(x, y, z)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S21BBF. | Class(es): C14 | Usage: $D = S21BBE(X, Y, Z, IFAIL)$ | On-line doc: CALL GAMSDOC S21BBE (or @PRT NAG*DOC.S21BBE) | Access: LIB NBS*NAG
- S21BBF** Elliptic integrals, symmetrised integral of 1st kind, $rf(x, y, z)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S21BBE. | Class(es): C14 | Usage: $D = S21BBF(X, Y, Z, IFAIL)$ | On-line doc: CALL GAMSDOC S21BBF (or @PRT NAG*DOC.S21BBF) | Access: LIB NBS*NAG
- S21BCE** Elliptic integrals, symmetrised integral of 2nd kind, $rd(x, y, z)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S21BCF. | Class(es): C14 | Usage: $D = S21BCE(X, Y, Z, IFAIL)$ | On-line doc: CALL GAMSDOC S21BCE

(or @PRT NAG*DOC.S21BCE) | Access: LIB NBS*NAG

S21BCF Elliptic integrals, symmetrised integral of 2nd kind, rd(x,y,z). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S21BCE. | Class(es): C14 | Usage: D = S21BCF (X, Y, Z, IFAIL) | On-line doc: CALL GAMSDOC S21BCF (or @PRT NAG*DOC.S21BCF) | Access: LIB NBS*NAG

S21BDE Elliptic integrals, symmetrised integral of 3rd kind, rj(x,y,z,r). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S21BDF. | Class(es): C14 | Usage: D = S21BDE (X, Y, Z, R, IFAIL) | On-line doc: CALL GAMSDOC S21BDE (or @PRT NAG*DOC.S21BDE) | Access: LIB NBS*NAG

S21BDF Elliptic integrals, symmetrised integral of 3rd kind, rj(x,y,z,r). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S21BDE. | Class(es): C14 | Usage: D = S21BDF (X, Y, Z, R, IFAIL) | On-line doc: CALL GAMSDOC S21BDF (or @PRT NAG*DOC.S21BDF) | Access: LIB NBS*NAG

SAMPLE Randomly selects without replacement values from one or more vectors, optionally carrying along other vectors. | Command in MINITAB Proprietary interactive system. Class(es): L6a10 | Usage: SAMPLE K rows from C [...],C], put into C [...],C] | On-line doc: HELP SAMPLE (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

SAMPP Computes the sample 100P percent point (where P is between 0.0 and 1.0, exclusively) of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1d | Usage: CALL SAMPP(X,N,P,IWRITE,PP) | On-line doc: CALL GAMSDOC SAMPP (or @PRT DATAPAC*DOC.SAMPP) | Access: LIB NBS*DAPAC

SASUM Compute single precision sum of absolute values of components of vector. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Double precision version is DASUM. | Class(es): D1a3a | Usage: S = SASUM(N,SX,INCX) | On-line doc: CALL GAMSDOC SASUM (or @PRT CMLIB*DOC.SASUM/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

SAXPY Compute a constant times a vector plus a vector, all single precision. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Double precision version is DAXPY. | Class(es): D1a7 | Usage: CALL SAXPY(N,SA,SX,INCX,SY,INCY) | On-line doc: CALL GAMSDOC SAXPY (or @PRT CMLIB*DOC.SAXPY/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

SCALE Computes 4 estimates of the scale (variation, scatter, dispersion) of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1b | Usage: CALL SCALE(X,N) | On-line doc: CALL GAMSDOC SCALE (or @PRT DATAPAC*DOC.SCALE) | Access: LIB NBS*DAPAC

SCASUM Compute complex sum of absolute values of components of vector. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. | Class(es): D1a3a | Usage: S = SCASUM(N,CX,INCX) | On-line doc: CALL GAMSDOC SCASUM (or @PRT CMLIB*DOC.SCASUM/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

SCHDC Compute Cholesky decomposition of real positive definite matrix with optional pivoting. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DCHDC. | Class(es): D2b1b | Usage: CALL SCHDC(A,LDA,P,WORK,JPVT,JOB,INFO) | On-line doc: CALL GAMSDOC SCHDC (or @PRT CMLIB*DOC.SCHDC/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB

SCHDD Downdates Cholesky factorization of real positive definite matrix. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DCHDD. | Class(es): D7b | Usage: CALL SCHDD(R,LDR,P,X,Z,LDZ,NZ,Y,RHO,C,S,INFO) | On-line doc: CALL GAMSDOC SCHDD (or @PRT CMLIB*DOC.SCHDD/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SCHDC

SCHEX Updates Cholesky factorization of real positive definite matrix. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DCHEX. | Class(es): D7b | Usage: CALL SCHEX(R,LDR,P,K,L,Z,LDZ,NZ,C,S,JOB) | On-line doc: CALL GAMSDOC SCHEX (or @PRT CMLIB*DOC.SCHEX/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SCHDC

SCHUD Updates Cholesky factorization of real positive definite matrix. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DCHUD. | Class(es): D7b | Usage: CALL SCHUD(R,LDR,P,X,Z,LDZ,NZ,Y,RHO,C,S) | On-line doc: CALL GAMSDOC SCHUD (or @PRT CMLIB*DOC.SCHUD/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SCHDC

SCNRM2 Compute the Euclidean length or L2 norm of a complex vector. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. | Class(es): D1a3b | Usage: S = SCNRM2(N,CX,INCX) | On-line doc: CALL GAMSDOC SCNRM2 (or @PRT CMLIB*DOC.SCNRM2/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

SCOPY Copy a vector X to a vector Y, both single precision. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Double precision version is DCOPY. | Class(es): D1a5 | Usage: CALL SCOPY(N,SX,INCX,SY,INCY) | On-line doc: CALL

- GAMSDOC SCOPY** (or @PRT CMLIB*DOC.SCOPY/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- SCOPYM** Copies negative of array SX into array SY, with corresponding increments INCX and INCY. | Portable single precision Fortran subprogram in XBLAS sublibrary of CMLIB library. | Class(es): D1a5 | Usage: CALL SCOPYM(N,SX,INCX,SY,INCY) | On-line doc: CALL GAMSDOC SCOPYM (or @PRT CMLIB*DOC.SCOPYM/XBLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/XBLAS | Access: LIB NBS*CMLIB
- SCOV** Calculates covariance matrix for a nonlinear dating fitting problem. This subprogram is intended to be used after a successful return from either of the subprograms SNLS1 or SNLS1E. | Portable single precision Fortran subprogram in SNLS1E sublibrary of CMLIB library. | Class(es): -0- | Usage: CALL SCOV(FCN,IOP,T,M,N,X,FVEC,R,LDR,INFO,WA1,WA2,WA3,WA4) | On-line doc: CALL GAMSDOC SCOV (or @PRT CMLIB*DOC.SCOV/SNLS1E) | Access: LIB NBS*CMLIB
- SD** Computes the sample standard deviation (with denominator N-1) of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1b | Usage: CALL SD(X,N,IWRITE,XSD) | On-line doc: CALL GAMSDOC SD (or @PRT DATAPAC*DOC.SD) | Access: LIB NBS*DAPAC
- SDASSL** Solves the system of differential/algebraic equations of the form $g(t, y, y_{\text{prime}}) = 0$, with given initial values. | Portable single precision Fortran subprogram in SDASSL sublibrary of CMLIB library. Double precision version is DDASSL. | Class(es): I1a1b | Usage: CALL SDASSL(RES,NEQ,T,Y,YPRIME,TOUT,INFO,RTOL,ATOL,IDID,RWORK,LRW,IWORK,LIW,RPAR,IPAR,JAC) | On-line doc: CALL GAMSDOC SDASSL (or @PRT CMLIB*DOC.SDASSL/SDASSL) | Tests: CMLIB*TEST-SOURCE.SDASSL/SDASSL | Access: LIB NBS*CMLIB
- SDOT** Compute single precision dot product. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Double precision version is DDOT. | Class(es): D1a4 | Usage: S = SDOT(N,SX,INCX,SY,INCY) | On-line doc: CALL GAMSDOC SDOT (or @PRT CMLIB*DOC.SDOT/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- SDRIV1** Numerical Integration, Initial Value Problems, Ordinary Differential Eqs., Gear Stiff Formulas, Easy to Use. | Portable single precision Fortran subprogram in SDRIV sublibrary of CMLIB library. Double precision version is DDRIV1. | Class(es): I1a2 I1a1b | Usage: CALL SDRIV1(N,T,Y,TOUT,MSTATE,EPS,WORK,LENW) | On-line doc: CALL GAMSDOC SDRIV1 (or @PRT CMLIB*DOC.SDRIV1/SDRIV) | Tests: CMLIB*TEST-SOURCE.\$F/SDRIV, CMLIB*TEST-SOURCE.\$Q/SDRIV | Access: LIB NBS*CMLIB
- SDRIV2** Numerical Integration, Initial Value Problems, Ordinary Differential Eqs., Gear / Adams Formulas. | Portable single precision Fortran subprogram in SDRIV sublibrary of CMLIB library. Double precision version is DDRIV2. | Class(es): I1a2 I1a1b | Usage: CALL SDRIV2(N,T,Y,F,TOUT,MSTATE,IROOT,EPS,EWT,MINT,WORK,LENW,IWORK,LENIW,G) | On-line doc: CALL GAMSDOC SDRIV2 (or @PRT CMLIB*DOC.SDRIV2/SDRIV) | Tests: CMLIB*TEST-SOURCE.\$F/SDRIV, CMLIB*TEST-SOURCE.\$Q/SDRIV | Access: LIB NBS*CMLIB
- SDRIV3** Numerical Integration, Initial Value Problems, Ordinary Differential Eqs., Implicit Eqs., Sparse Jacobians. | Portable single precision Fortran subprogram in SDRIV sublibrary of CMLIB library. Double precision version is DDRIV3. | Class(es): I1a2 I1a1b | Usage: CALL SDRIV3(N,T,Y,F,NSTATE,TOUT,NTASK,IROOT,EPS,EWT,IERROR,MINT,MITER,IMPL,ML,MU,MXORD,HMAX,WORK,LENW,IWORK,LENIW,JACOB,N,FA,NDE,MXSTEP,G) | On-line doc: CALL GAMSDOC SDRIV3 (or @PRT CMLIB*DOC.SDRIV3/SDRIV) | Tests: CMLIB*TEST-SOURCE.\$F/SDRIV, CMLIB*TEST-SOURCE.\$Q/SDRIV | Access: LIB NBS*CMLIB
- SDRV** Solves sparse symmetric systems of linear algebraic equations by Gaussian elimination without pivoting. | Portable single precision Fortran subprogram in YSMP sublibrary of CMLIB library. | Class(es): D2b4 | Usage: CALL SDRV(N,P,IP,IA,JA,A,B,Z,NSP,ISP,RSP,ESP,PATH,FLAG) | On-line doc: CALL GAMSDOC SDRV (or @PRT CMLIB*DOC.SDRV/YSMP) | Tests: CMLIB*TEST-SOURCE.\$Q1/YSMP | Access: LIB NBS*CMLIB | See also: ODRV
- SDSDOT** Compute single precision dot product and add a constant using double precision accumulation. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. | Class(es): D1a4 | Usage: S = SDSDOT(N,SB,SX,INCX,SY,INCY) | On-line doc: CALL GAMSDOC SDSDOT (or @PRT CMLIB*DOC.SDSDOT/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- SEPELI** Solves separable elliptic boundary value problems on a rectangle. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): I2b1a2 | Usage: CALL SEPELI(INTL,IORDER,A,B,M,MBDCND,BDA,ALPHA,BDB,BETA,C,D,N,NBDCND,BDC,GAMA,BDD,XNU,COFX,COFY,GRHS,USOL,IDMN,W,PERTRB,IERROR) | On-line doc: CALL GAMSDOC SEPELI (or @PRT CMLIB*DOC.SEPELI/FSHPK) | Tests: CMLIB*TEST-SOURCE.SEPELI/FSHPK | Access: LIB NBS*CMLIB
- SEPX4** Solves separable elliptic boundary value problems on a rectangle with constant coefficients in one direction. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): I2b1a2 | Usage: CALL SEPX4(IORDER,A,B,M,MBDCND,BDA,ALPHA,BDB,BETA,C,D,N,NBDCND,BDC,BDD,COFX,GRHS,USOL,IDMN,W,PERTRB,IERROR) | On-line doc: CALL GAMSDOC SEPX4 (or @PRT CMLIB*DOC.SEPX4/FSHPK) | Tests: CMLIB*TEST-SOURCE.SEPX4/FSHPK | Access: LIB NBS*CMLIB
- SET** Create a constant vector or a vector of integers in increments of 1 or more or with other patterns. | Command in MINITAB Proprietary interactive system. | Class(es): L2 | Usage: SET [K repetitions of] (K,...,K) [repeated K times] into column C | On-line doc: HELP SET (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

- SETC** Set a specified number of values in a complex array equal to a constant. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): D1a1 | Usage: CALL SETC (N,V,B) | On-line doc: CALL GAMSDOC SETC (or @PRT PORT*DOC.SETC) | Access: LIB NBS*PORT
- SETD** Set a specified number of values in a double precision array equal to a constant. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is SETR. | Class(es): D1a1 | Usage: CALL SETD (N,V,B) | On-line doc: CALL GAMSDOC SETD (or @PRT PORT*DOC.SETD) | Access: LIB NBS*PORT
- SETERR** Sets the error indicator and depending on options prints a message and provides a dump for PORT library programs. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): R3c | Usage: CALL SETERR (MESSG,NMESSG,NERR,IOP) | On-line doc: CALL GAMSDOC SETERR (or @PRT PORT*DOC.SETERR) | Access: LIB NBS*PORT
- SETI** Set a specified number of values in an integer array equal to a constant. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): D1a1 | Usage: CALL SETI (N,V,B) | On-line doc: CALL GAMSDOC SETI (or @PRT PORT*DOC.SETI) | Access: LIB NBS*PORT
- SETL** Set a specified number of values in a logical array equal to a constant. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): D1a1 | Usage: CALL SETL (N,V,B) | On-line doc: CALL GAMSDOC SETL (or @PRT PORT*DOC.SETL) | Access: LIB NBS*PORT
- SETR** Set a specified number of values in a real array equal to a constant. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is SETD. | Class(es): D1a1 | Usage: CALL SETR (N,V,B) | On-line doc: CALL GAMSDOC SETR (or @PRT PORT*DOC.SETR) | Access: LIB NBS*PORT
- SGBCO** Computes LU factorization of real band matrix and estimates its condition. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DGBCO. | Class(es): D2a2 | Usage: CALL SGBCO(ABD,LDA,N,ML,MU,IPVT,RCOND,Z) | On-line doc: CALL GAMSDOC SGBCO (or @PRT CMLIB*DOC.SGBCO/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB
- SGBDI** Uses LU factorization of real band matrix to compute its determinant. (No provision for computing matrix inverse.). | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DGBDI. | Class(es): D3a2 | Usage: CALL SGBDI(ABD,LDA,N,ML,MU,IPVT,DET) | On-line doc: CALL GAMSDOC SGBDI (or @PRT CMLIB*DOC.SGBDI/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SGBCO,SGBFA
- SGBFA** Computes LU factorization of real band matrix. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DGBFA. | Class(es): D2a2 | Usage: CALL SGBFA(ABD,LDA,N,ML,MU,IPVT,INFO) | On-line doc: CALL GAMSDOC SGBFA (or @PRT CMLIB*DOC.SGBFA/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB
- SGBSL** Uses LU factorization of real band matrix to solve systems. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DGBSL. | Class(es): D2a2 | Usage: CALL SGBSL(ABD,LDA,N,ML,MU,IPVT,B,JOB) | On-line doc: CALL GAMSDOC SGBSL (or @PRT CMLIB*DOC.SGBSL/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SGBCO,SGBFA
- SGECO** Computes LU factorization of real general matrix and estimates its condition. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DGEEO. | Class(es): D2a1 | Usage: CALL SGECO(A,LDA,N,IPVT,RCOND,Z) | On-line doc: CALL GAMSDOC SGECO (or @PRT CMLIB*DOC.SGECO/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB
- SGEDI** Uses LU factorization of real general matrix to compute its determinant and/or inverse. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DGEDI. | Class(es): D2a1 D3a1 | Usage: CALL SGEDI(A,LDA,N,IPVT,DET,WORK,JOB) | On-line doc: CALL GAMSDOC SGEDI (or @PRT CMLIB*DOC.SGEDI/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SGECO,SGEFA
- SGEEV** Computes the eigenvalues and, optionally, the eigenvectors of a general real matrix. | Portable single precision Fortran subprogram in LICEPACK sublibrary of CMLIB library. | Class(es): D4a2 | Usage: CALL SGEEV(A,LDA,N,E,V,LDV,WORK,JOB,INFO) | On-line doc: CALL GAMSDOC SGEEV (or @PRT CMLIB*DOC.SGEEV/LICEPACK) | Tests: CMLIB*TEST-SOURCE.SGEEV/LICEPACK | Access: LIB NBS*CMLIB
- SGEFA** Computes LU factorization of real general matrix. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DGEFA. | Class(es): D2a1 | Usage: CALL SGEFA(A,LDA,N,IPVT,INFO) | On-line doc: CALL GAMSDOC SGEFA (or @PRT CMLIB*DOC.SGEFA/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB
- SGEFS** Factors and solves a general NXN single precision system of linear equations. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. Double precision version is DGEFS. | Class(es): D2a1 | Usage: CALL SGEFS(A,LDA,N,V,ITASK,IND,WORK,IWORK) | On-line doc: CALL GAMSDOC SGEFS (or @PRT

CMLIB*DOC.SGEFS/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F1/LINDRIVES | Access: LIB NBS*CMLIB

SGEIR Factors and solves a general single precision system of linear equations and estimates solution accuracy (requires NxN extra storage). | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. | Class(es): D2a1 | Usage: CALL SGEIR(A,LDA,N,V,ITASK,IND,WORK,IWORK) | On-line doc: CALL GAMSDOC SGEIR (or @PRT CMLIB*DOC.SGEIR/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F1/LINDRIVES | Access: LIB NBS*CMLIB

SGESL Uses LU factorization of real general matrix to solve systems. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DGESL. | Class(es): D2a1 | Usage: CALL SGESL(A,LDA,N,IPVT,B,JOB) | On-line doc: CALL GAMSDOC SGESL (or @PRT CMLIB*DOC.SGESL/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SGECON,SGEFA

SGLSS Solves linear least squares problems. Emphasis is put on detecting possible rank deficiency. Performs QR factorization using Householder transformations. Easy-to-use driver for LLSIA and ULSIA. | Portable single precision Fortran subprogram in SGLSS sublibrary of CMLIB library. | Class(es): D0 | Usage: CALL SGLSS(A,MDA,M,N,B,MDB,NB,RNORM,WORK,LW,IWORK,LIW,INFO) | On-line doc: CALL GAMSDOC SGLSS (or @PRT CMLIB*DOC.SGLSS/SGLSS) | Tests: CMLIB*TEST-SOURCE.SGLSS/SGLSS | Access: LIB NBS*CMLIB

SGTSL Factors a real tridiagonal matrix and simultaneously solves a system. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DGTSL. | Class(es): D2a2a | Usage: CALL SGTSL(N,C,D,E,B,INFO) | On-line doc: CALL GAMSDOC SGTSL (or @PRT CMLIB*DOC.SGTSL/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB

SICIEI Computes sine, cosine, exponential integral as well as hyperbolic sin, cosine, exponential integral. | Portable double precision Fortran subprogram in STEGUN sublibrary of MATHWARE library. | Class(es): C5 C6 | Usage: CALL SICIEI(IC,X,SI,CI,CH,EI,EXNEI,SHI,CHI,CHII,IERR) | On-line doc: @PRT,S MATHWARE*STEGUN.SICIEI/DOC | Access: See individual sublibrary documentation

SIN Sine of x. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4a | Usage: Y = SIN (X) | On-line doc: CALL GAMSDOC SIN (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

SINDG Computes the Sine of an argument in degrees. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DSINDG. | Class(es): C4a | Usage: R=SINDG(X) | On-line doc: CALL GAMSDOC SINDG (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

SINH Computes hyperbolic sin sinh(x). | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DSINH. | Class(es): C4c | Usage: X = SINH (X) | On-line doc: CALL GAMSDOC SINH (or @PRT PORT*DOC.SINH) | Access: LIB NBS*PORT

SINQB Computes (fast) Fourier transform of quarter wave data. Backward (fast) sine transform. Performs Fourier synthesis. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): J1a3 | Usage: CALL SINQB(N,X,WSAVE) | On-line doc: CALL GAMSDOC SINQB (or @PRT CMLIB*DOC.SINQB/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: SINQF,SINQI

SINQF Computes (fast) Fourier transform of quarter wave data. Forward (fast) sine transform. Performs Fourier analysis. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): J1a3 | Usage: CALL SINQF(N,X,WSAVE) | On-line doc: CALL GAMSDOC SINQF (or @PRT CMLIB*DOC.SINQF/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: SINQB,SINQI

SINQI Initialize array WSAVE for SUBROUTINE SINQF and SINQB. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): -0- | Usage: CALL SINQI(N,WSAVE) | On-line doc: CALL GAMSDOC SINQI (or @PRT CMLIB*DOC.SINQI/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: SINQF,SINQB

SINT Computes (fast) Fourier sine transform of an odd sequence X(I). | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): J1a3 | Usage: CALL SINT(N,X,WSAVE) | On-line doc: CALL GAMSDOC SINT (or @PRT CMLIB*DOC.SINT/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: SINTI

SINTI Initialize array WSAVE for SUBROUTINE SINT. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): -0- | Usage: CALL SINTI(N,WSAVE) | On-line doc: CALL GAMSDOC SINTI (or @PRT CMLIB*DOC.SINTI/FFTPKG) | Tests: CMLIB*TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS*CMLIB | See also: SINT

SKIPR Reads through (skips over) NLHEAD lines from input unit = 5. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): N1 | Usage: CALL SKIPR(NLHEAD) | On-line doc: CALL GAMSDOC SKIPR (or @PRT DATAPAC*DOC.SKIPR) | Access: LIB NBS*DAPAC

SLIP1 Solves a fully implicit difference scheme for a one-dimensional system of parabolic differential equations with general boundary conditions on a specified equidistant spatial grid using a specified order spatial method. Order and stepsize in time and error estimate are computed. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2a1a | Usage: CALL SLIP1 (TOL, IERROR, IGLOBAL, IDOKU, IOUT, DTDR, ISTR, TANF, TEND, HANF, IIMAX, HMIN, NV, NX, NOX, NA, NF, X, UII, U, UT, UTTEST, UX, UXX, DU, DUT, DUX, DUXX, F, DFGLT, DFGLTU, DFGLTO, DFGLX, DFXMAX, TAU, E, FU, FUT, FUX, FUXX,Q,MV, MNX,MOX,INTOK,LOUT) | On-line doc: CALL GAMSDOC SLIP1 (or @PRT SLDGL*DOC.SLIP1) | Tests:

- SL1P2** Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with initial conditions determined from the solution of an ode boundary value problem. User specifies equidistant grid and order in space, stepsize and order in time and error estimate computed. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2a1a | Usage: CALL SL1P2 (TOL, IERROR, IGLOB, IDOKU, IOUT, DTDR, ISTR, TANF, TEND, HANF, HMAX, HMIN, NV, NX, NOX, NA, NF, X, UH, U, UT, UTFEST, UX, UXX, DU, DUT, DUX, DUXX, F, DFGLT, DFGLTU, DFGLTO, DFGLX, DFXMAX, TAU, E, FU, FUT, FUX, FUXX, Q, MV, MNX, MOX, INTOK, LOUT) | On-line doc: CALL GAMSDOC SL1P2 (or @PRT SLDGL*DOC.SL1P2) | Tests: SLDGL*TEST-SOURCE.SL1P2,SLDGL*TEST-DATA.SL1P2 | Access: LIB NBS*SLDGL
- SL1P3** Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with specified initial and boundary conditions. For a given relative accuracy an equidistant spatial grid and optimal order are computed as well as stepsize and order in time. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2a1a | Usage: CALL SL1P3 (TOL, IAN, IERROR, IGLOB, IDOKU, IOUT, DTDR, ISTR, TANF, TEND, HANF, HMAX, HMIN, NV, NX, NOX, NA, NF, X, XAN, XEN, UH, U, UT, UTFEST, UX, UXX, DU, DUT, DUX, DUXX, F, DFGLT, DFGLTU, DFGLTO, DFGLX, DFXMAX, TAU, E, FU, FUT, FUX, FUXX, Q, MV, MNX, MOX, INTOK, LOUT) | On-line doc: CALL GAMSDOC SL1P3 (or @PRT SLDGL*DOC.SL1P3) | Tests: SLDGL*TEST-SOURCE.SL1P3,SLDGL*TEST-DATA.SL1P3 | Access: LIB NBS*SLDGL
- SL1P4** Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with initial conditions determined from the solution of an ode boundary value problem. Given a relative accuracy an equidistant spatial grid and optimal order are determined. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2a1a | Usage: CALL SL1P4 (TOL, IAN, IERROR, IGLOB, IDOKU, IOUT, DTDR, ISTR, TANF, TEND, HANF, HMAX, HMIN, NV, NX, NOX, NA, NF, X, XAN, XEN, UH, U, UT, UTFEST, UX, UXX, DU, DUT, DUX, DUXX, F, DFGLT, DFGLTU, DFGLTO, DFGLX, DFXMAX, TAU, E, FU, FUT, FUX, FUXX, Q, MV, MNX, MOX, INTOK, LOUT) | On-line doc: CALL GAMSDOC SL1P4 (or @PRT SLDGL*DOC.SL1P4) | Tests: SLDGL*TEST-SOURCE.SL1P4,SLDGL*TEST-DATA.SL1P4 | Access: LIB NBS*SLDGL
- SL1P5** Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with specified initial and boundary conditions and non-equidistant spatial grid and spatial order. Stepsize and order in time are computed as well as an estimate of global error. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2a1a | Usage: CALL SL1P5 (TOL, IERROR, IGLOB, IDOKU, IOUT, DTDR, ISTR, TANF, TEND, HANF, HMAX, HMIN, NV, NX, NOX, NA, ABX, ABXX, FABX, FABXX, X, XS, Z1, Z2, STX, UH, U, UT, UTFEST, UX, UXX, DU, DUT, DUX, DUXX, F, DFGLT, DFGLTU, DFGLTO, DFGLX, DFXMAX, TAU, E, FU, FUT, FUX, FUXX, Q, MV, MNX, MOX, INTOK, LOUT) | On-line doc: CALL GAMSDOC SL1P5 (or @PRT SLDGL*DOC.SL1P5) | Tests: SLDGL*TEST-SOURCE.SL1P5,SLDGL*TEST-DATA.SL1P5 | Access: LIB NBS*SLDGL
- SL1P6** Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with initial conditions determined from an ode boundary value problem. A non-equidistant spatial grid and spatial order are specified and the stepsize and order in time and error estimate are computed. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2a1a | Usage: CALL SL1P6 (TOL, IERROR, IGLOB, IDOKU, IOUT, DTDR, ISTR, TANF, TEND, HANF, HMAX, HMIN, NV, NX, NOX, NA, ABX, ABXX, FABX, FABXX, X, XS, Z1, Z2, STX, UH, U, UT, UTFEST, UX, UXX, DU, DUT, DUX, DUXX, F, DFGLT, DFGLTU, DFGLTO, DFGLX, DFXMAX, TAU, E, FU, FUT, FUX, FUXX, Q, MV, MNX, MOX, INTOK, LOUT) | On-line doc: CALL GAMSDOC SL1P6 (or @PRT SLDGL*DOC.SL1P6) | Tests: SLDGL*TEST-SOURCE.SL1P6,SLDGL*TEST-DATA.SL1P6 | Access: LIB NBS*SLDGL
- SL1P7** Solves a fully implicit difference scheme for a system of one-dimensional parabolic equations with specified initial and boundary conditions. For a specified relative accuracy a non-equidistant spatial grid and optimal order are computed as well as an estimate of global error. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2a1a | Usage: CALL SL1P7 (TOL, IAN, IERROR, IGLOB, IDOKU, IOUT, DTDR, ISTR, TANF, TEND, HANF, HMAX, HMIN, NV, NX, NOX, NA, ABX, ABXX, FABX, FABXX, X, XS, Z1, Z2, STX, XAN, XEN, UH, U, UT, UTFEST, UX, UXX, DU, DUT, DUX, DUXX, F, DFGLT, DFGLTU, DFGLTO, DFGLX, DFXMAX, TAU, E, FU, FUT, FUX, FUXX, Q, MV, MNX, MOX, INTOK, LOUT) | On-line doc: CALL GAMSDOC SL1P7 (or @PRT SLDGL*DOC.SL1P7) | Tests: SLDGL*TEST-SOURCE.SL1P7,SLDGL*TEST-DATA.SL1P7 | Access: LIB NBS*SLDGL
- SL1P8** Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with initial conditions determined from an ode boundary value problem. For a given relative accuracy a non-equidistant spatial grid and order and error estimate are computed. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2a1a | Usage: CALL SL1P8 (TOL, IAN, IERROR, IGLOB, IDOKU, IOUT, DTDR, ISTR, TANF, TEND, HANF, HMAX, HMIN, NV, NX, NOX, NA, ABX, ABXX, FABX, FABXX, X, XS, Z1, Z2, STX, XAN, XEN, UH, U, UT, UTFEST, UX, UXX, DU, DUT, DUX, DUXX, F, DFGLT, DFGLTU, DFGLTO, DFGLX, DFXMAX, TAU, E, FU, FUT, FUX, FUXX, Q, MV, MNX, MOX, INTOK, LOUT) | On-line doc: CALL GAMSDOC SL1P8 (or @PRT SLDGL*DOC.SL1P8) | Tests: SLDGL*TEST-SOURCE.SL1P8,SLDGL*TEST-DATA.SL1P8 | Access: LIB NBS*SLDGL
- SL2E2** Solves by a fully implicit difference scheme a two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. For a prescribed relative accuracy the gridpoints and order of the method are automatically determined. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2b2 | Usage: CALL SL2E2 (MV, MV2, MNX, MNY, MOX, ABX, ABY, ABXX, ABYY, DF1, DF2, DU, DUH, FABX, FABY, FABXX, FABYY, FU, FUX, FUY, FUXX, FUY, ST, UMAX, UMIN, UH, MAXIT) | On-line doc: CALL GAMSDOC SL2E2 (or @PRT SLDGL*DOC.SL2E2) | Tests: SLDGL*TEST-SOURCE.SL2E2,SLDGL*TEST-DATA.SL2E2 | Access: LIB NBS*SLDGL
- SL2E4** Solves by a fully implicit difference scheme a two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. A non-equidistant grid and the order of the method are prescribed. An estimate of the discretization error is computed. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2b2 | Usage: CALL SL2E4 (MAXIT, MOX, MV, MV2, MNX, MNY, ABX, ABY, ABXX, ABYY, DF1, DF2, DU, DUH, FABX, FABY, FABXX, FABYY, FU, FUX, FUY, FUXX, FUY, ST, UH, UMAX, UMIN) | On-line doc: CALL GAMSDOC SL2E4 (or @PRT SLDGL*DOC.SL2E4) | Tests: SLDGL*TEST-SOURCE.SL2E4,SLDGL*TEST-DATA.SL2E4 | Access: LIB NBS*SLDGL

- SL2E5** Solves by a fully implicit difference scheme a two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. For a prescribed relative accuracy the non-equidistant grid and order of the method are determined automatically. Discretization error estimated. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): 12b2 | Usage: CALL SL2E5 (MV,MV2,MNX,MNY,MOX, ABX,ABY,ABXX, ABYY,DF1,DF2,DU, DUH,FABX,FABY, FABXX,FABYY,FU,FUX,FUY, FUXX,FUYF,ST,UMAX,UMIN,UH, MAXIT) | On-line doc: CALL GAMSDOC SL2E5 (or @PRT SLDGL*DOC.SL2E5) | Tests: SLDGL*TEST-SOURCE.SL2E5,SLDGL*TEST-DATA.SL2E5 | Access: LIB NBS*SLDGL
- SL2E6** Solves by a fully implicit difference scheme a two-dimensional elliptic boundary value problem with general boundary conditions on a rectangle. For a prescribed equidistant grid and relative accuracy the routine automatically determines the order of the method. Error estimate provided. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): 12b2 | Usage: CALL SL2E6 (MAXIT,MOX,MV,MV2,MNX,MNY,ABS, ABY,ABXX,ABYY,DF1,DF2, DU,DUH, FABX,FABY, FABXX,FABYY,FU,UX,FUY,FUXX,FUYF, ST,UXH,UMAX,UMIN) | On-line doc: CALL GAMSDOC SL2E6 (or @PRT SLDGL*DOC.SL2E6) | Tests: SLDGL*TEST-SOURCE.SL2E6,SLDGL*TEST-DATA.SL2E6 | Access: LIB NBS*SLDGL
- SL2E7** Solves a fully implicit difference scheme for two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. A non-equidistant grid and the order of the method are prescribed. An estimate of the discretization error is provided. Line iteration used. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): 12b2 | Usage: CALL SL2E7 (MAXIT,MOX,MV,MV2,MNX,MNY, ABX,ABY,ABXX,ABYY, DU1,DU2,DUX, Q1,Q2, FABX,FABY, FABXX,FABYY,FU,FUX,FUY, FUXX,FUYF,ST,UH,UMAX,UMIN, DUA,DEF,M1,M2) | On-line doc: CALL GAMSDOC SL2E7 (or @PRT SLDGL*DOC.SL2E7) | Tests: SLDGL*TEST-SOURCE.SL2E7,SLDGL*TEST-DATA.SL2E7 | Access: LIB NBS*SLDGL
- SL2E8** Solves a fully implicit difference scheme for a two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. For a prescribed relative accuracy the gridpoints and order of the method are determined. Approximation solved by line iteration. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): 12b2 | Usage: CALL SL2E8 (MAXIT,MOX,MV, MNX,MNY,ABX,ABY,ABXX, ABYY,DU1,DU2,DUX, Q1,Q2, FABX,FABY,FABXX, FABYY,FU,FUX,FUY,FUXX,FUYF,ST, UH,UMAX,UMIN,DUA, DEF,M1,M2) | On-line doc: CALL GAMSDOC SL2E8 (or @PRT SLDGL*DOC.SL2E8) | Tests: SLDGL*TEST-SOURCE.SL2E8,SLDGL*TEST-DATA.SL2E8 | Access: LIB NBS*SLDGL
- SL2EB1** Solves a fully implicit difference scheme for a system of two-dimensional elliptic equations on a general region with Dirichlet or Neumann boundary conditions specified. A non-equidistant grid and the order of the method are prescribed. An estimate of the discretization error is provided. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): 12b2 | Usage: CALL SL2EB1 (MAXIT,MV,MN,MNX,MNY, MG,IGITT,UH, UMAX,UMIN,DFGLX, DFGLY, DU,DUX,FU,FUX,FUY,FUXX, FUYF,X,Y,RPX, RPY,ALPHAX,ALPHAY, NALX, NALY,IFL, INABS,INREL,IFEHL) | On-line doc: CALL GAMSDOC SL2EB1 (or @PRT SLDGL*DOC.SL2EB1) | Tests: SLDGL*TEST-SOURCE.SL2EB1,SLDGL*TEST-DATA.SL2EB1 | Access: LIB NBS*SLDGL
- SL2P1** Solves a fully implicit difference scheme for an implicit system of two-dimensional parabolic equations on a rectangle with specified initial and boundary conditions. User provides non-equidistant spatial grid and spatial orders and stepsize and order in time and error estimate are computed. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): 12a1b | Usage: CALL SL2P1 (ISTART,MAXINT,MV,MV2,MNX, MNY,MOX,ABX, ABY,ABXX,ABYY, UH,DU,DUX, DFGL, DFGLX,DFGLY,FABX,FABY, FABXX,FABYY,FU,FUT, FUX,FUY,FUXX, FUYF,STX,UMAX,UMIN) | On-line doc: CALL GAMSDOC SL2P1 (or @PRT SLDGL*DOC.SL2P1) | Tests: SLDGL*TEST-SOURCE.SL2P1,SLDGL*TEST-DATA.SL2P1 | Access: LIB NBS*SLDGL
- SL2P2** Solves a fully implicit difference scheme for an implicit system of two-dimensional parabolic equations. Initial conditions are computed from an elliptic boundary value problem. User specifies non-equidistant spatial grid and order and stepsize and order in time and error est. are computed. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): 12a1b | Usage: CALL SL2P2 (ISTART,MAXINT, MV,MV2,MNX,MNY,MOX, ABX,ABY,ABXX,ABYY, UH,DU,DUX, DFGL,DFGLX,DFGLY, FABX,FABY,FABXX,FABYY,FU, FUT,FUX,FUY,FUXX, FUYF,STX,UMAX,MIN) | On-line doc: CALL GAMSDOC SL2P2 (or @PRT SLDGL*DOC.SL2P2) | Tests: SLDGL*TEST-SOURCE.SL2P2,SLDGL*TEST-DATA.SL2P2 | Access: LIB NBS*SLDGL
- SL2P3** Solves a fully implicit difference scheme for an implicit system of parabolic equations with given initial and boundary conditions. For a given error tolerance an optimal combination of spatial gridpoint distribution and order are computed as well as an estimate of the global error. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): 12a1b | Usage: CALL SL2P3 (ISTART,MAXINT,MV,M2, MNX,MNY,MOX,ABX, ABY,ABXX,ABYY, UH,DU,DUX, DFGL,DFGLX,DFGLY,FABX, FABY,FABXX,FABYY,FU,FUT, FUX,FUY,FUXX, FUYF,STX,UMAX,UMIN) | On-line doc: CALL GAMSDOC SL2P3 (or @PRT SLDGL*DOC.SL2P3) | Tests: SLDGL*TEST-SOURCE.SL2P3,SLDGL*TEST-DATA.SL2P3 | Access: LIB NBS*SLDGL
- SL2P4** Solves a fully implicit difference scheme for an implicit system of two-dimensional parabolic equations with the initial conditions determined as the solution of an elliptic boundary value problem. For a given tolerance optimal spatial gridpoint distribution and order are computed. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): 12a1b | Usage: CALL SL2P4 (ISTART,MAXINT,MV,MV2, MNX,MNY,MOX,ABX, ABY,ABXX,ABYY, UH,DU,DUX, DFGL, DFGLX,DFGLY,FABX,FABY, FABXX,FABYY,FU,FUT, FUX,FUY,FUXX, FUYF,STX,UMAX,UMIN) | On-line doc: CALL GAMSDOC SL2P4 (or @PRT SLDGL*DOC.SL2P4) | Tests: SLDGL*TEST-SOURCE.SL2P4,SLDGL*TEST-DATA.SL2P4 | Access: LIB NBS*SLDGL
- SL3E1** Solves a fully implicit difference scheme for a system of three-dimensional elliptic equations on a parallelepiped with general boundary conditions. A non-equidistant grid and the order of the method are prescribed. An estimate of the discretization error is computed. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): 12b2 | Usage: CALL SL3E1 (MAXIT,MOX,MV,MNX, MNY,MNZ,ABX, ABY,ABZ,ABXX,ABYY, ABZZ,DU,DUALT, DUX,FABX,FABY, FABZ,FABXX,FABYY, FABZZ,FU,FUX,FUY,FUZ, FUXX,FUYF, FUZZ,ST,UH,UMAX,UMIN) | On-line doc: CALL GAMSDOC SL3E1 (or @PRT SLDGL*DOC.SL3E1) | Tests: SLDGL*TEST-SOURCE.SL3E1,SLDGL*TEST-DATA.SL3E1 | Access: LIB NBS*SLDGL

- SL3P1** Solves a fully implicit difference scheme for a three-dimensional system of parabolic equations with specified initial and boundary conditions on a parallelepiped. The user provides a non-equidistant spatial grid and spatial orders and stepsize and order in time and an error estimate are computed. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2a1b | Usage: CALL SL3P1 (ISTART,MAXINT,MV,MNX, MNY,MNZ,MOX, ABX,ABY,ABZ,ABXX, ABYY,ABZZ,UH, DU,DUX,DFGL,DUALT, FABX,FABY,FABZ,FABXX, FABYY,FABZZ,FU, FUT,FUX, FUY,FUZ, FUXX,FUYX,FUZZ, STX,UMAX,UMIN) | On-line doc: CALL GAMSDOC SL3P1 (or @PRT SLDGL*DOC.SL3P1) | Tests: SLDGL*TEST-SOURCE.SL3P1,SLDGL*TEST-DATA.SL3P1 | Access: LIB NBS*SLDGL
- SLGA1** Solves by difference methods an initial value problem for a system of ordinary differential equations. The stepsize and order of the method are chosen automatically to maintain prescribed local discretization error bounds. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I1a1b | Usage: CALL SLGA1 (N,NMAX,TOL,REL, IABS,IGLOB,IOUT, IDOKU,DXDR,HANF, HMAX,XANF,XEND,YO,Y, YSTR,YSFEST,YJACOB,DY,DEFEKT, GLFEHL,ALOFEH, UNFEHL,OBFEHL,E,EN, TAU,GEW,LOUT) | On-line doc: CALL GAMSDOC SLGA1 (or @PRT SLDGL*DOC.SLGA1) | Tests: SLDGL*TEST-SOURCE.SLGA1,SLDGL*TEST-DATA.SLGA1 | Access: LIB NBS*SLDGL
- SLGA2** Solves by difference methods a mixed implicit/algebraic initial value problem for a system of ordinary differential equations. Stepsize and order are chosen automatically to maintain estimated local discretization error within prescribed bounds. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I1a1b | Usage: CALL SLGA2 (N,NMAX,TOL,REL, IABS,IGLOB,IOUT, IDOKU,DXDR,HANF, HMAX,XANF,XEND,YO,Y, YSTR,YSFEST, FY,FYX,YJACOB,DY, DEFEKT, GLFEHL, ALOFEH, UNFEHL,OBFEHL, E,EN,TAU,GEW,LOUT) | On-line doc: CALL GAMSDOC SLGA2 (or @PRT SLDGL*DOC.SLGA2) | Tests: SLDGL*TEST-SOURCE.SLGA2,SLDGL*TEST-DATA.SLGA2 | Access: LIB NBS*SLDGL
- SLGA3** Solves by difference methods an initial value problem for an explicit system of ordinary differential equations. Chooses stepsize and order of method to maintain estimate of local discretization error within prescribed bounds. Also provides values of first derivative at output points. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I1a1b | Usage: CALL SLGA3 (N,NMAX,TOL, REL,IABS,IGLOB, IOUT,IDOKU,DXDR,ISTR, HANF, HMAX,XANF,XEND,YO,Y, YSTR,YSFEST,YJACOB,DY, DEFEKT,GLFEHL, ALOFEH, UNFEHL,OBFEHL, E,EN,TAU,GEW,LOUT) | On-line doc: CALL GAMSDOC SLGA3 (or @PRT SLDGL*DOC.SLGA3) | Tests: SLDGL*TEST-SOURCE.SLGA3,SLDGL*TEST-DATA.SLGA3 | Access: LIB NBS*SLDGL
- SLGA4** Solves by difference methods a mixed algebraic/implicit initial value problem for a first order system of ordinary differential equations. chooses stepsize and order to maintain estimate of local discretization error within prescribed bounds. Also provides derivative at output points. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I1a1b | Usage: CALL SLGA4 (N,NMAX,TOL,REL,IABS, IGLOB,IOUT, IDOKU,DXDR,ISTR, HANF,HMAX, XANF,XEND,YO,Y, YSTR,YSFEST,FY,FYX,YJACOB, DY,DEFEKT,GLFEHL, ALOFEH, UNFEHL,OBFEHL,E,EN, TAU,GEW,LOUT) | On-line doc: CALL GAMSDOC SLGA4 (or @PRT SLDGL*DOC.SLGA4) | Tests: SLDGL*TEST-SOURCE.SLGA4,SLDGL*TEST-DATA.SLGA4 | Access: LIB NBS*SLDGL
- SLGR0** Solves by difference methods on a non-equidistant grid a second-order implicit system of ordinary differential equations with implicit boundary conditions prescribed. An estimate of the discretization error is provided. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I1b2 | Usage: CALL SLGR0 (IDOKU,NX,NOX,NV, MNX,MOX,MV,NA,KONV, ABX,ABXX,DFGL, DY,DYX,DYXX, F,FABX, FABXX,FY,FYX,FYXX, Q,RUNDF,ST,X,XS,Y, YX,YXX,Z1,Z2, LOUT) | On-line doc: CALL GAMSDOC SLGR0 (or @PRT SLDGL*DOC.SLGR0) | Tests: SLDGL*TEST-SOURCE.SLGR0,SLDGL*TEST-DATA.SLGR0 | Access: LIB NBS*SLDGL
- SLGR1** Solves by difference methods on an equidistant grid an implicit system of second-order ordinary differential equations with prescribed implicit boundary conditions. An estimate of the discretization error is provided. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I1b2 | Usage: CALL SLGR1 (IDOKU,NX,NOX,NV,MNX,MOX,MV, NA,KONV,IX,NF,ABX,ABXX, DFGL,DY,DYX, DYXX,F, FABX,FABXX,FY,FYX,FYXX,Q, RUNDF,ST,X,Y,YX,YXX,ZW, LOUT) | On-line doc: CALL GAMSDOC SLGR1 (or @PRT SLDGL*DOC.SLGR1) | Tests: SLDGL*TEST-SOURCE.SLGR1,SLDGL*TEST-DATA.SLGR1 | Access: LIB NBS*SLDGL
- SLGR2** Solves by difference methods on a non-equidistant grid an implicit system of second-order ordinary differential equations with implicit boundary values. For a given relative accuracy gridpoints and order of method are chosen automatically to minimize number of gridpoints. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I1b2 | Usage: CALL SLGR2 (IAN,IDOKU,NA,NOX, NXE,NX,NV, MOX,MNX,MV,ABX,ABXX, DFGL,DY,DYX,DYXX, F,FABX,FABXX, FY,FYX,FYXX,H,Q, RUNDF,ST,TOL,X,XAN,XEN, XNEU,XS,Y, YNEU,YX,YXX,Z1,Z2,LOUT) | On-line doc: CALL GAMSDOC SLGR2 (or @PRT SLDGL*DOC.SLGR2) | Tests: SLDGL*TEST-SOURCE.SLGR2,SLDGL*TEST-DATA.SLGR2 | Access: LIB NBS*SLDGL
- SLGR3** Solves by difference methods on an equidistant grid an implicit system of second-order ordinary differential equations with implicit boundary values. For a prescribed relative accuracy the order of the method is chosen to minimize the number of gridpoints. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I1b2 | Usage: CALL SLGR3 (IAN,IDOKU,NA,NOX,NXE,NX,NV, MOX,MNX,MV,IX,NF,ABX, ABXX,DFGL,DY, DYX,DYXX,F,FABX,FABXX,FY,FYX, FYXX,Q,RUNDF,ST,TOL,X,SAN, XEN,XNEU, Y,YX,YXX,ZW,LOUT) | On-line doc: CALL GAMSDOC SLGR3 (or @PRT SLDGL*DOC.SLGR3) | Tests: SLDGL*TEST-SOURCE.SLGR3,SLDGL*TEST-DATA.SLGR3 | Access: LIB NBS*SLDGL
- SLGR4** Solves by difference methods on a non-equidistant grid an implicit system of second-order ordinary differential equations with implicit boundary conditions. For a prescribed relative accuracy the gridpoints and order of the method are chosen to minimize the number of gridpoints. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I1b2 | Usage: CALL SLGR4 (IAN,IDOKU, IFILE,IF1,LOPT, LOUT,MOX,MNX,MV,MXX,NA, NOX,NV,NX, ABX,ABXX,DFGL, DY,DYX,DYXX,F,FABX, FABXX,FY,FYX,FYXX,H,Q, RUNDF, ST,TOL,X,XAN,XEN, XNEU,XS,Y,YNEU, YX,YXX,ZF1,ZF2,C) | On-line doc: CALL GAMSDOC SLGR4 (or @PRT SLDGL*DOC.SLGR4) | Tests: SLDGL*TEST-SOURCE.SLGR4,SLDGL*TEST-DATA.SLGR4 | Access: LIB NBS*SLDGL

- SLGR5** Solves by difference methods on a non-equidistant grid an implicit system of second-order ordinary differential equations with implicit boundary conditions. For a given relative accuracy the gridpoints and order of method are chosen to minimize number of gridpts. Suited for boundary layer problems. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): 11b2 | Usage: CALL SLGR5 (IAN, IDOKU, IF1, IF2, LOPT, LOUT, MOX, MNX, MV, MXX, NA, NOX, NV, NX, ABX, ABXX, DFGL, DY, DYX, DYXX, F, FABX, FABXX, FY, FYX, FYXX, H, Q, RUND, ST, TOL, X, XAN, XEN, XNEU, XS, Y, YNEU, YX, YXX, ZF1, ZF2, C) | On-line doc: CALL GAMSDOC SLGR5 (or @PRT SLDGL*DOC.SLGR5) | Tests: SLDGL*TEST-SOURCE.SLGR5, SLDGL*TEST-DATA.SLGR5 | Access: LIB NBS*SLDGL
- SLVBLK** Solves $Ax=b$ where A is an almost block diagonal matrix. These arise in finite element or piecewise polynomial approximation. | Portable single precision Fortran subprogram in SLVBLK sublibrary of CMLIB library. | Class(es): D2a4 | Usage: CALL SLVBLK (BLOKS, INTEGS, NBLOKS, B, IPIVOT, X, IFLAG) | On-line doc: CALL GAMSDOC SLVBLK (or @PRT CMLIB*DOC.SLVBLK/SLVBLK) | Tests: CMLIB*TEST-SOURCE.\$Q/SLVBLK | Access: LIB NBS*CMLIB
- SMONOD** Test if a double precision vector is strictly monotone increasing or decreasing. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is SMONOR. | Class(es): R2 | Usage: L = SMONOD (X, N, INC) | On-line doc: CALL GAMSDOC SMONOD (or @PRT PORT*DOC.SMONOD) | Access: LIB NBS*PORT
- SMONOI** Test if an integer vector is strictly monotone increasing or decreasing. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): R2 | Usage: L = SMONOI (X, N, INC) | On-line doc: CALL GAMSDOC SMONOI (or @PRT PORT*DOC.SMONOI) | Access: LIB NBS*PORT
- SMONOR** Test if a real vector is strictly monotone increasing or decreasing. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is SMONOD. | Class(es): R2 | Usage: L = SMONOR (X, N, INC) | On-line doc: CALL GAMSDOC SMONOR (or @PRT PORT*DOC.SMONOR) | Access: LIB NBS*PORT
- SMSNO** Minimize a general unconstrained objective function using finite difference gradients and secant Hessian approximations. | Portable single precision Fortran subprogram in NL2SN sublibrary of CMLIB library. Double precision version is DSMSNO. | Class(es): G1b1a | Usage: CALL SMSNO (N, D, X, CALCF < IV, LIV, LV, V, UIPARM, URPARN, UFPARM) | On-line doc: CALL GAMSDOC SMSNO (or @PRT CMLIB*DOC.SMSNO/NL2SN) | Tests: CMLIB*TEST-SOURCE.\$F2/NL2SN, CMLIB*TEST-SOURCE.\$Q2/NL2SN | Access: LIB NBS*CMLIB
- SNBCO** Factors a real band matrix by Gaussian elimination and estimates condition of the matrix. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. Double precision version is DNBCO. | Class(es): D2a2 | Usage: CALL SNBCO (ABE, LDA, N, ML, MU, IPVT, RCOND, Z) | On-line doc: CALL GAMSDOC SNBCO (or @PRT CMLIB*DOC.SNBCO/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F1/LINDRIVES | Access: LIB NBS*CMLIB
- SNBDI** Computes the determinant of a single precision band matrix using factors previously computed. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. Double precision version is DNBDI. | Class(es): D3a2 | Usage: CALL SNBDI (ABE, LDA, N, ML, MU, IPVT, DET) | On-line doc: CALL GAMSDOC SNBDI (or @PRT CMLIB*DOC.SNBDI/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F1/LINDRIVES | Access: LIB NBS*CMLIB | See also: SNBCO, SNBFA
- SNBFA** Factors a single precision band matrix by elimination. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. Double precision version is DNBFA. | Class(es): D2a2 | Usage: CALL SNBFA (ABE, LDA, N, ML, MU, IPVT, INFO) | On-line doc: CALL GAMSDOC SNBFA (or @PRT CMLIB*DOC.SNBFA/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F1/LINDRIVES | Access: LIB NBS*CMLIB
- SNBFS** Factors and solves a general nonsymmetric single precision banded system of linear equations. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. Double precision version is DNBFS. | Class(es): D2a2 | Usage: CALL SNBFS (ABE, LDA, N, ML, MU, V, ITASK, IND, WORK, IWORK) | On-line doc: CALL GAMSDOC SNBFS (or @PRT CMLIB*DOC.SNBFS/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F1/LINDRIVES | Access: LIB NBS*CMLIB
- SNBIR** Factors and solves a general nonsymmetric single precision banded system of equations and estimates solution accuracy (needs $N \times (2ML + MU)$ extra storage). | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. | Class(es): D2a2 | Usage: CALL SNBIR (ABE, LDA, N, ML, MU, V, ITASK, IND, WORK, IWORK) | On-line doc: CALL GAMSDOC SNBIR (or @PRT CMLIB*DOC.SNBIR/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F1/LINDRIVES | Access: LIB NBS*CMLIB
- SNBSL** Solves a general nonsymmetric single precision banded system of linear equations using factors computed previously. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. Double precision version is DNBSL. | Class(es): D2a2 | Usage: CALL SNBSL (ABE, LDA, N, ML, MU, IPVT, B, JOB) | On-line doc: CALL GAMSDOC SNBSL (or @PRT CMLIB*DOC.SNBSL/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F1/LINDRIVES | Access: LIB NBS*CMLIB | See also: SNBCO, SNBFA
- SNLS1** Minimizes the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm. Flexible usage, including various options for providing Jacobian. Covariance matrix is available via the subroutine SCOV. | Portable single precision Fortran subprogram in SNLS1E sublibrary of CMLIB library. | Class(es): K1b1a1 K1b1a2 | Usage: CALL SNLS1 (FCN, IOPT, M, N, X, FVEC, FJAC, LDFJAC, FTOL, XTOL, GTOL, MAXFEV, EPSFCN, DIAG, MODE, FACTOR, NPRINT, INFO, NFEV, NJEV, IPVT, QTF, WA1, WA2, WA3, WA4) | On-line doc: CALL GAMSDOC SNLS1 (or @PRT CMLIB*DOC.SNLS1/SNLS1E) | Tests: CMLIB*TEST-SOURCE.\$Q/SNLS1E | Access: LIB NBS*CMLIB | See also: SCOV, CHIKDER checks user's Jacobian routine if desired

- SNLS1E** Minimizes the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm. An easy to use driver for SNLS1. The covariance matrix is available by calling the subroutine SCOV. | Portable single precision Fortran subprogram in SNLS1E sublibrary of CMLIB library. | Class(es): K1b1a1 K1b1a2 | Usage: CALL SNLS1E(FCN,IOPT,M,N,X,FVEC,TOL,NPRINT,INFO,IW,WA,LWA) | On-line doc: CALL GAMSDOC SNLS1E (or @PRT CMLIB*DOC.SNLS1E/SNLS1E) | Tests: CMLIB*TEST-SOURCE.\$Q/SNLS1E | Access: LIB NBS*CMLIB | See also: SCOV, CHKDER checks user's Jacobian routine if desired
- SNRM2** Compute the Euclidean length or L2 norm of a single precision vector. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Double precision version is DNRM2. | Class(es): D1a3b | Usage: S = SNRM2(N,SX,INCX) | On-line doc: CALL GAMSDOC SNRM2 (or @PRT CMLIB*DOC.SNRM2/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- SNRM2** Finds the length (Euclidean norm) of a vector, without underflow or overflow. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DNRM2. | Class(es): D1a3b | Usage: X = SNRM2 (N, X, INCX) | On-line doc: CALL GAMSDOC SNRM2 (or @PRT PORT*DOC.SNRM2) | Access: LIB NBS*PORT
- SNSQ** Finds a zero of a system of N nonlinear equations in N variables by a modification of the Powell hybrid method. Flexible usage. | Portable single precision Fortran subprogram in SNLS1E sublibrary of CMLIB library. | Class(es): F2a | Usage: CALL SNSQ(FCN,JAC,IOPT,N,X,FVEC,FJAC,LDFJAC,XTOL,MAXFEV,ML,MU, EPSFCN, DIAG,MODE,FACTOR,NPRINT,INFO,NFEV,NJEV,R,LR,QTF,WA1,WA2,WA3, WA4) | On-line doc: CALL GAMSDOC SNSQ (or @PRT CMLIB*DOC.SNSQ/SNLS1E) | Tests: CMLIB*TEST-SOURCE.\$Q/SNLS1E | Access: LIB NBS*CMLIB | See also: CHKDER checks user's Jacobian routine if desired.
- SNSQE** Finds a zero of a system of N nonlinear equations in N variables by a modification of Powell's hybrid method. An easy to use driver for SNSQ. | Portable single precision Fortran subprogram in SNLS1E sublibrary of CMLIB library. | Class(es): F2a | Usage: CALL SNSQE(FCN,JAC,IOPT,N,X,FVEC,TOL,NPRINT,INFO,WA,LWA) | On-line doc: CALL GAMSDOC SNSQE (or @PRT CMLIB*DOC.SNSQE/SNLS1E) | Tests: CMLIB*TEST-SOURCE.\$Q/SNLS1E | Access: LIB NBS*CMLIB | See also: CHKDER if user supplies Jacobian this checks for consistency.
- SODS** Solves an overdetermined system of linear equations. For full rank matrices the unique least squares solution is provided. The least squares solution of minimal length can be obtained in the rank deficient case. | Portable single precision Fortran subprogram in SUDSSODS sublibrary of CMLIB library. | Class(es): D0 | Usage: CALL SODS(A,X,B,NEQ,NUK,NRDA,IFLAG,WORK,IWORK) | On-line doc: CALL GAMSDOC SODS (or @PRT CMLIB*DOC.SODS/SUDSSODS) | Tests: CMLIB*TEST-SOURCE.SODS/SUDSSODS | Access: LIB NBS*CMLIB
- SOR** Iterative solution of large sparse systems of linear equations. SOR method, adaptive parameter selection. | Portable single precision Fortran subprogram in ITPACK sublibrary of MATHWARE library. | Class(es): D2b4 12b4b D2a4 | Usage: CALL SOR(N,IA,JA,A,RHS,U,IWKSP,NW,WKSP,IPARM,RPARM,IER) | On-line doc: @PRT,S MATHWARE*ITPACK.DOCUMENT | Access: See individual sublibrary documentation
- SORT** Sorts (in ascending order) the N elements of the vector X and puts the resulting N sorted values into the vector Y. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): N6a2b1 | Usage: CALL SORT(X,N,Y) | On-line doc: CALL GAMSDOC SORT (or @PRT DATAPAC*DOC.SORT) | Access: LIB NBS*DAPAC
- SORT** Sorts a vector in ascending order and optionally carries along other vectors. | Command in MINITAB Proprietary interactive system. Class(es): N6a2b1 | Usage: SORT the values in C [carry along corresponding rows of C,..., C] put into C [corresponding rows into C,...,C] | On-line doc: HELP SORT (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- SORTC** Sorts (in ascending order) the N elements of the vector X and rearranges the elements of the vector Y. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): N6a2b1 | Usage: CALL SORTC(X,Y,N,XS,YC) | On-line doc: CALL GAMSDOC SORTC (or @PRT DATAPAC*DOC.SORTC) | Access: LIB NBS*DAPAC
- SORTP** Sorts (in ascending order) the N elements of the vector X, puts the resulting N sorted values into the vector Y, and puts the position (in the original vector X) of each of the sorted values into the single precision vector XPOS. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): N6a2b1 | Usage: CALL SORTP(X,N,Y,XPOS) | On-line doc: CALL GAMSDOC SORTP (or @PRT DATAPAC*DOC.SORTP) | Access: LIB NBS*DAPAC
- SOS** Finds a zero of a system of N nonlinear equations in N unknowns using Brown's method. | Portable single precision Fortran subprogram in SNLS1E sublibrary of CMLIB library. | Class(es): F2a | Usage: CALL SOS(FNC,NEQ,X,RTOLX,ATOLX,TOLF,IFLAG,RW,LRW,IW,LIW) | On-line doc: CALL GAMSDOC SOS (or @PRT CMLIB*DOC.SOS/SNLS1E) | Tests: CMLIB*TEST-SOURCE.\$Q/SNLS1E | Access: LIB NBS*CMLIB
- SPBCO** Uses Cholesky algorithm to compute factorization of real positive definite band matrix and estimates its condition. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPBCO. | Class(es): D2b2 | Usage: CALL SPBCO(ABD,LDA,N,M,RCOND,Z,INFO) | On-line doc: CALL GAMSDOC SPBCO (or @PRT CMLIB*DOC.SPBCO/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB
- SPBDI** Uses factorization of real positive definite band matrix to compute its determinant. (No provision for matrix inverse.). | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPBDI. | Class(es): D3b2 | Usage: CALL SPBDI(ABD,LDA,N,M,DET) | On-line doc: CALL GAMSDOC SPBDI (or @PRT CMLIB*DOC.SPBDI/LINPACKS) |

Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SPBCO SPBFA

- SPBFA** Uses Cholesky algorithm to compute factorization of real positive definite band matrix. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPBFA. | Class(es): D2b2 | Usage: CALL SPBFA(ABD,LDA,N,M,INFO) | On-line doc: CALL GAMSDOC SPBFA (or @PRT CMLIB*DOC.SPBFA/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB
- SPBSL** Uses factorization of real positive definite band matrix to solve systems. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPBSL. | Class(es): D2b2 | Usage: CALL SPBSL(ABD,LDA,N,M,B) | On-line doc: CALL GAMSDOC SPBSL (or @PRT CMLIB*DOC.SPBSL/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SPBCO SPBFA
- SPCORR** Computes the Spearman rank correlation coefficient between the two sets of data in the input vectors X and Y. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L4b1b L1e1b | Usage: CALL SPCORR(X,Y,N,IWRITE,SPC) | On-line doc: CALL GAMSDOC SPCORR (or @PRT DATAPAC*DOC.SPCORR) | Access: LIB NBS*DAPAC
- SPECAN** An interactive spectral analysis package for time series data. Produces periodograms, cumulative periodograms, continuous Fourier power spectra, cumulative power spectra, Fourier amplitude spectra, maximum entropy spectra, and integrated maximum entropy spectra. With DISSPLA graphics. | Portable stand-alone program using SPECTRLAN command language. | Class(es): L10f | On-line doc: CALL GAMSDOC SPECAN (or @PRT SPECTRLAN*DOC.SPECAN) | Access: CALL SCD*CTSLIB.SPECAN (in CTS)
- SPENC** Spence Dilogarithm, $= - \int_0^x ((\ln \text{ of abs. val. of } 1-y) / y) dy$. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DSPENC. | Class(es): C5 | Usage: Y = SPENC (X) | On-line doc: CALL GAMSDOC SPENC (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB
- SPLN1** Evaluates a function and derivatives described previously by an expansion in terms of B-splines. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DSPLN1. | Class(es): E3 K6 | Usage: CALL SPLN1 (K,T,N,A,X,NX,ID,NID,FX) | On-line doc: CALL GAMSDOC SPLN1 (or @PRT PORT*DOC.SPLN1) | Access: LIB NBS*PORT
- SPLN2** Evaluates a function described by a previously determined expansion in B-splines. More flexible than SPLN1. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DSPLN2. | Class(es): E3 K6 | Usage: CALL SPLN2 (K,T,N,A,X,NX,ID,NID,FX,IDIM,ADIFF,ILO,ILEFT) | On-line doc: CALL GAMSDOC SPLN2 (or @PRT PORT*DOC.SPLN2) | Access: LIB NBS*PORT
- SPLND** Evaluates at a given set of points a function described by a previously determined expansion in terms of B-splines. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DSPLND. | Class(es): E3 K6 | Usage: CALL SPLND (K,T,N,A,X,NX,MD,FX) | On-line doc: CALL GAMSDOC SPLND (or @PRT PORT*DOC.SPLND) | Access: LIB NBS*PORT
- SPLNE** Evaluates at a set of points, a function described by a previously determined expansion in terms of B-splines. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DSPLNE. | Class(es): E3 K6 | Usage: CALL SPLNE (K,T,N,A,X,NX,FX) | On-line doc: CALL GAMSDOC SPLNE (or @PRT PORT*DOC.SPLNE) | Access: LIB NBS*PORT
- SPLNI** Integrates a function described previously by an expansion in terms of B-splines. Several integrations can be performed in one call. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DSPLNI. | Class(es): H2a2a1 E3 K6 | Usage: CALL SPLNI (K,T,N,A,X,NX,FX) | On-line doc: CALL GAMSDOC SPLNI (or @PRT PORT*DOC.SPLNI) | Access: LIB NBS*PORT
- SPLP** Solves linear optimization problems, that is, it minimizes the linear function (TRANSPPOSE OF COSTS)*X subject to $A*X=W$, where the entries of the vectors X and W may have simple upper or lower bounds. Uses a sparse storage mode for the matrix A and out-of-core scratch storage. | Portable single precision Fortran subprogram in SPLP sublibrary of CMLIB library. | Class(es): G2a2 | Usage: CALL SPLP(USRMAT,MRELAS,NVARS,COSTS,PRGOPT,DATTRY,BL,BU,IND, INFO,PRIMAL, DUALS, IBASIS,WORK,LW,IWORK,LIW) | On-line doc: CALL GAMSDOC SPLP (or @PRT CMLIB*DOC.SPLP/SPLP) | Tests: CMLIB*TEST-SOURCE.\$Q/SPLP, CMLIB*TEST-SOURCE.\$F/SPLP | Access: LIB NBS*CMLIB
- SPLT** Displays a 50x100 character line printer scatter plot with user control of the plotting symbol used for each point. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c2 Q1 | Usage: CALL SPLT (Y, X, N, ISYM) | On-line doc: CALL GAMSDOC SPLT (or @PRT STATLIB*DOC.SPLT) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB
- SPLTH** Displays a 50x50 character line printer scatter plot with user control of the plotting symbol used for each point. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c2 Q1 | Usage: CALL SPLTH (Y, X, N, ISYM) | On-line doc: CALL GAMSDOC SPLTH (or @PRT STATLIB*DOC.SPLTH) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB
- SPLTHL** Displays a 50x50 character line printer scatter plot with user control of the plot limits and of the plotting symbol used for each point. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c2 Q1 | Usage: CALL SPLTHL (Y, X, N, ISYM, NOUT, YLB, YUB, XLB, XUB) | On-line doc: CALL GAMSDOC SPLTHL (or @PRT STATLIB*DOC.SPLTHL) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB
- SPLTL** Displays a 50x100 character line printer scatter plot with user control of the plot limits and of the plotting symbol used for each point. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c2 Q1 | Usage: CALL SPLTL (Y, X, N, ISYM, NOUT, YLB, YUB, XLB, XUB) | On-line doc: CALL GAMSDOC SPLTL (or @PRT STATLIB*DOC.SPLTL) | Tests: STATLIB*TEST.DEMO1

| Access: LIB NBS*STATLIB

- SPOCO** Uses Cholesky algorithm to factor real positive definite matrix and estimate its condition. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPOCO. | Class(es): D2b1b | Usage: CALL SPOCO(A,LDA,N,RCOND,Z,INFO) | On-line doc: CALL GAMSDOC SPOCO (or @PRT CMLIB*DOC.SPOCO/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB
- SPODI** Uses factorization of real positive definite matrix to compute its determinant and/or inverse. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPODI. | Class(es): D2b1b D3b1b | Usage: CALL SPODI(A,LDA,N,DET,JOB) | On-line doc: CALL GAMSDOC SPODI (or @PRT CMLIB*DOC.SPODI/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SPOCO SPOFA
- SPOFA** Uses Cholesky algorithm to factor real positive definite matrix. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPOFA. | Class(es): D2b1b | Usage: CALL SPOFA(A,LDA,N,INFO) | On-line doc: CALL GAMSDOC SPOFA (or @PRT CMLIB*DOC.SPOFA/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB
- SPOFS** Factors and solves a symmetric positive definite single precision system of linear equations. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. Double precision version is DPOFS. | Class(es): D2b1b | Usage: CALL SPOFS(A,LDA,N,V,ITASK,IND,WORK) | On-line doc: CALL GAMSDOC SPOFS (or @PRT CMLIB*DOC.SPOFS/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F1/LINDRIVES | Access: LIB NBS*CMLIB
- SPOIR** Factors and solves a symmetric positive definite single precision system of equations and estimates solution accuracy (needs NxN extra storage). | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. | Class(es): D2b1b | Usage: CALL SPOIR(A,LDA,N,V,ITASK,IND,WORK) | On-line doc: CALL GAMSDOC SPOIR (or @PRT CMLIB*DOC.SPOIR/LINDRIVES) | Tests: CMLIB*TEST-SOURCE.\$F1/LINDRIVES | Access: LIB NBS*CMLIB
- SPOSL** Uses factorization of real positive definite matrix to solve systems. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPOSL. | Class(es): D2b1b | Usage: CALL SPOSL(A,LDA,N,B) | On-line doc: CALL GAMSDOC SPOSL (or @PRT CMLIB*DOC.SPOSL/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SPOCO SPOFA
- SPPCO** Uses Cholesky algorithm to factor real positive definite matrix stored in packed form and estimate its condition. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPPCO. | Class(es): D2b1b | Usage: CALL SPPCO(AP,N,RCOND,Z,INFO) | On-line doc: CALL GAMSDOC SPPCO (or @PRT CMLIB*DOC.SPPCO/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB
- SPPDI** Uses factorization of real positive definite matrix stored in packed form to compute its determinant and/or inverse. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPPDI. | Class(es): D2b1b D3b1b | Usage: CALL SPPDI(AP,N,DET,JOB) | On-line doc: CALL GAMSDOC SPPDI (or @PRT CMLIB*DOC.SPPDI/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SPPCO SPPFA
- SPPFA** Uses Cholesky algorithm to factor real positive definite matrix stored in packed form. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPPFA. | Class(es): D2b1b | Usage: CALL SPPFA(AP,N,INFO) | On-line doc: CALL GAMSDOC SPPFA (or @PRT CMLIB*DOC.SPPFA/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB
- SPPSL** Uses factorization of real positive definite matrix stored in packed form to solve systems. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPPSL. | Class(es): D2b1b | Usage: CALL SPPSL(AP,N,B) | On-line doc: CALL GAMSDOC SPPSL (or @PRT CMLIB*DOC.SPPSL/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SPPCO SPPFA
- SPTSL** Decomposes real symmetric positive definite tridiagonal matrix and simultaneously solves a system. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPTSL. | Class(es): D2b2a | Usage: CALL SPTSL(N,D,E,B) | On-line doc: CALL GAMSDOC SPTSL (or @PRT CMLIB*DOC.SPTSL/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB
- SQRANK** For solving linear systems in least squares sense. Computes the QR decomposition of matrix using LINPACK subroutines. | Portable single precision Fortran subprogram in SQRSS sublibrary of CMLIB library. Double precision version is DQRANK. | Class(es): D5 | Usage: CALL SQRANK(A,LDA,M,N,TOL,KR,JPVT,QRAUX,WORK) | On-line doc: CALL GAMSDOC SQRANK (or @PRT CMLIB*DOC.SUMMARY/SQRSS and CMLIB*DOC.SQRANK/SQRSS) | Tests: CMLIB*TEST-SOURCE.\$F/SQRSS, CMLIB*TEST-SOURCE.\$Q/SQRSS | Access: LIB NBS*CMLIB
- SQRDC** Computes QR decomposition of real general matrix. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DQRDC. | Class(es): D5 | Usage: CALL SQRDC(X,LDX,N,P,QRAUX,JPVT,WORK,JOB)

| On-line doc: CALL GAMSDOC SQRDC (or @PRT CMLIB*DOC.SQRDC/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB

SQRLSS For solving linear systems in least squares sense. Finds solution and residual after matrix factored by SQRANK. | Portable single precision Fortran subprogram in SQRLSS sublibrary of CMLIB library. Double precision version is DQRLSS. | Class(es): D9 | Usage: CALL SQRLSS(A,LDA,M,N,KR,B,X,RSD,JPVT,QRAUX) | On-line doc: CALL GAMSDOC SQRLSS (or @PRT CMLIB*DOC.SUMMARY/SQRLSS and CMLIB*DOC.SQRLSS/SQRLSS) | Tests: CMLIB*TEST-SOURCE.\$F/SQRLSS, CMLIB*TEST-SOURCE.\$Q/SQRLSS | Access: LIB NBS*CMLIB | See also: SQRANK

SQRSL Applies the output of SQRDC to compute coordinate transformations, projections, and least squares solutions (general real matrix). | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DQSL. | Class(es): D9 D2a1 | Usage: CALL SQRSL(X,LDX,N,K,QRAUX,Y,QY,QTY,B,RSD,XB,JOB,INFO) | On-line doc: CALL GAMSDOC SQRSL (or @PRT CMLIB*DOC.SQRSL/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SQRDC

SQRT Square root. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C2 | Usage: Y = SQRT(X) | On-line doc: CALL GAMSDOC SQRT (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

SROT Apply Givens plane rotation to a single precision vector. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Double precision version is DROT. | Class(es): D1a8 | Usage: CALL SROT(N,SX,INCX,SY,INCY,SC,SS) | On-line doc: CALL GAMSDOC SROT (or @PRT CMLIB*DOC.SROT/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

SROTG Construct Givens plane rotation of single precision matrix. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Double precision version is DROTG. | Class(es): D1b10 | Usage: CALL SROTG(SA,SB,SC,SS) | On-line doc: CALL GAMSDOC SROTG (or @PRT CMLIB*DOC.SROTG/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

SROTM Apply modified Givens plane rotation to single precision vector. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Double precision version is DROTM. | Class(es): D1a8 | Usage: CALL SROTM(N,SX,INCX,SY,INCY,SPARAM) | On-line doc: CALL GAMSDOC SROTM (or @PRT CMLIB*DOC.SROTM/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

SROTMG Construct modified Givens plane rotation of single precision matrix. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Double precision version is DROTMG. | Class(es): D1b10 | Usage: CALL SROTMG(SD1,SD2,SB1,SB2,SPARAM) | On-line doc: CALL GAMSDOC SROTMG (or @PRT CMLIB*DOC.SROTMG/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

SRTAD Actively sorts double precision data into ascending order. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is SRTAR. | Class(es): N6a2b2 | Usage: CALL SRTAD (A,INT,N) | On-line doc: CALL GAMSDOC SRTAD (or @PRT PORT*DOC.SRTAD) | Access: LIB NBS*PORT

SRTAH Actively sorts Hollerith data into ascending order. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): N6a2c | Usage: CALL SRTAH (A,L,INC,N) | On-line doc: CALL GAMSDOC SRTAH (or @PRT PORT*DOC.SRTAH) | Access: LIB NBS*PORT

SRTAI Actively sorts integer data into ascending order. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): N6a2a | Usage: CALL SRTAI (A,INT,N) | On-line doc: CALL GAMSDOC SRTAI (or @PRT PORT*DOC.SRTAI) | Access: LIB NBS*PORT

SRTAR Actively sorts real data into ascending order. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is SRTAD. | Class(es): N6a2b1 | Usage: CALL SRTAR (A,INT,N) | On-line doc: CALL GAMSDOC SRTAR (or @PRT PORT*DOC.SRTAR) | Access: LIB NBS*PORT

SRTDD Actively sorts double precision data into descending order. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is SRTDR. | Class(es): N6a2b2 | Usage: CALL SRTDD (A,INT,N) | On-line doc: CALL GAMSDOC SRTDD (or @PRT PORT*DOC.SRTDD) | Access: LIB NBS*PORT

SRTDH Actively sorts Hollerith data into descending order. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): N6a2c | Usage: CALL SRTDH (A,L,INC,N) | On-line doc: CALL GAMSDOC SRTDH (or @PRT PORT*DOC.SRTDH) | Access: LIB NBS*PORT

SRTDI Actively sorts integer data into descending order. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): N6a2a | Usage: CALL SRTDI (A,INT,N) | On-line doc: CALL GAMSDOC SRTDI (or @PRT PORT*DOC.SRTDI) | Access: LIB NBS*PORT

SRTDR Actively sorts real data into descending order. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is SRTDD. | Class(es): N6a2b1 | Usage: CALL SRTDR (A,INT,N) | On-line doc: CALL GAMSDOC SRTDR (or @PRT PORT*DOC.SRTDR) | Access: LIB NBS*PORT

SRTPAD Passively sorts double precision data into ascending order. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is SRTPAR. | Class(es): N6a1b2 | Usage: CALL SRTPAD (A,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTPAD (or @PRT PORT*DOC.SRTPAD) | Access: LIB NBS*PORT | See also: SRTDR

- SRTPAH** Passively sorts Hollerith data into ascending order. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): N6a1c | Usage: CALL SRTPAH (A,L,INC,N) | On-line doc: CALL GAMSDOC SRTPAH (or @PRT PORT*DOC.SRTPAH) | Access: LIB NBS*PORT | See also: SRTRH
- SRTPAI** Passively sorts integer data into ascending order. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): N6a1a | Usage: CALL SRTPAI (A,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTPAI (or @PRT PORT*DOC.SRTPAI) | Access: LIB NBS*PORT | See also: SRTRI
- SRTPAR** Passively sorts real data into ascending order. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is SRTPAD. | Class(es): N6a1b1 | Usage: CALL SRTPAR (A,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTPAR (or @PRT PORT*DOC.SRTPAR) | Access: LIB NBS*PORT | See also: SRTRR
- SRTPDD** Passively sorts double precision data into descending order. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is SRTPDR. | Class(es): N6a1b2 | Usage: CALL SRTPDD (A,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTPDD (or @PRT PORT*DOC.SRTPDD) | Access: LIB NBS*PORT | See also: SRTRD
- SRTPDH** Passively sorts Hollerith data into descending order. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): N6a1c | Usage: CALL SRTPDH (A,L,INC,N) | On-line doc: CALL GAMSDOC SRTPDH (or @PRT PORT*DOC.SRTPDH) | Access: LIB NBS*PORT | See also: SRTRH
- SRTPDI** Passively sorts integer data into descending order. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): N6a1a | Usage: CALL SRTPDI (A,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTPDI (or @PRT PORT*DOC.SRTPDI) | Access: LIB NBS*PORT | See also: SRTRI
- SRTPDR** Passively sorts real data into descending order. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is SRTPDD. | Class(es): N6a1b1 | Usage: CALL SRTPDR (A,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTPDR (or @PRT PORT*DOC.SRTPDR) | Access: LIB NBS*PORT | See also: SRTRR
- SRTRD** Rearranges double precision data according to permutation stored in IP. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is SRTRR. | Class(es): N8 | Usage: CALL SRTRD (A,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTRD (or @PRT PORT*DOC.SRTRD) | Access: LIB NBS*PORT
- SRTRH** Rearranges Hollerith data according to permutation stored in IP. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): N8 | Usage: CALL SRTRH (A,L,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTRH (or @PRT PORT*DOC.SRTRH) | Access: LIB NBS*PORT
- SRTRI** Rearranges integer data according to permutation stored in IP. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): N8 | Usage: CALL SRTRI (A,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTRI (or @PRT PORT*DOC.SRTRI) | Access: LIB NBS*PORT
- SRTRR** Rearranges real data according to permutation stored in IP. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is SRTRD. | Class(es): N8 | Usage: CALL SRTRR (A,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTRR (or @PRT PORT*DOC.SRTRR) | Access: LIB NBS*PORT
- SSCAL** Compute a constant times a vector, both single precision. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Double precision version is DSCAL. | Class(es): D1a6 | Usage: CALL SSCAL(N,SA,SX,INCX) | On-line doc: CALL GAMSDOC SSCAL (or @PRT CMLIB*DOC.SSCAL/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB
- SSICO** Computes factorization of real symmetric indefinite matrix and estimates its condition. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DSICO. | Class(es): D2b1a | Usage: CALL SSICO(A,LDA,N,KPVT,RCOND,Z) | On-line doc: CALL GAMSDOC SSICO (or @PRT CMLIB*DOC.SSICO/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB
- SSIDI** Uses factorization of real symmetric indefinite matrix to compute its determinant and/or inverse. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DSIDI. | Class(es): D2b1a D3b1a | Usage: CALL SSIDI(A,LDA,N,KPVT,DET,INERT,WORK,JOB) | On-line doc: CALL GAMSDOC SSIDI (or @PRT CMLIB*DOC.SSIDI/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SSICO SSIFA
- SSIEV** Computes the eigenvalues and, optionally, the eigenvectors of a real symmetric matrix. | Portable single precision Fortran subprogram in LICEPACK sublibrary of CMLIB library. | Class(es): D4a1 | Usage: CALL SSIEV(A,LDA,N,E,WORK,JOB,INFO) | On-line doc: CALL GAMSDOC SSIEV (or @PRT CMLIB*DOC.SSIEV/LICEPACK) | Tests: CMLIB*TEST-SOURCE.SSIEV/LICEPACK | Access: LIB NBS*CMLIB
- SSIFA** Computes factorization of real symmetric indefinite matrix. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DSIFA. | Class(es): D2b1a | Usage: CALL SSIFA(A,LDA,N,KPVT,INFO) | On-line doc: CALL GAMSDOC SSIFA (or @PRT CMLIB*DOC.SSIFA/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB

- SSISL** Uses factorization of real symmetric indefinite matrix to solve systems. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DSISL. | Class(es): D2b1a | Usage: CALL SSISL(A,LDA,N,KPVT,B) | On-line doc: CALL GAMSDOC SSISL (or @PRT CMLIB*DOC.SSISL/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SSICO SSIFA
- SSORCG** Iterative solution of large sparse systems of linear equations. SSOR method, conjugate gradient acceleration, adaptive parameter selection. | Portable single precision Fortran subprogram in ITPACK sublibrary of MATHWARE library. | Class(es): D2b4 I2b4b D2a4 | Usage: CALL SSORCG(N,IA,JA,A,RHS,U,IWKSP,NW,WKSP,IPARM,RPARM,IER) | On-line doc: @PRT,S MATHWARE*ITPACK.DOCUMENT | Access: See individual sublibrary documentation
- SSORSI** Iterative solution of large sparse systems of linear equations. SSOR method, Chebyshev acceleration, adaptive parameter selection. | Portable single precision Fortran subprogram in ITPACK sublibrary of MATHWARE library. | Class(es): D2b4 I2b4b D2a4 | Usage: CALL SSORSI(N,IA,JA,A,RHS,U,IWKSP,NW,WKSP,IPARM,RPARM,IER) | On-line doc: @PRT,S MATHWARE*ITPACK.DOCUMENT | Access: See individual sublibrary documentation
- SSORT** Sorts an array X (of N real numbers) into increasing or decreasing order. An optional array Y is carried along with X. | Portable single precision Fortran subprogram in SSORT sublibrary of CMLIB library. | Class(es): N6a2b1 | Usage: CALL SSORT(X,Y,N,KFLAG) | On-line doc: CALL GAMSDOC SSORT (or @PRT CMLIB*DOC.SSORT/SSORT) | Tests: CMLIB*TEST-SOURCE.\$Q/SSORT | Access: LIB NBS*CMLIB
- SSPAND** Simple random sampling with proportion data - inferences regarding the population proportion and total. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2c L4c | Usage: CALL SSPAND (IOPT,NBR,ALPHA,STAT,IDIST,IER) | On-line doc: CALL GAMSDOC SSPAND (or @PRT IMSL*DOC.SSPAND) | Access: LIB NBS*IMSL
- SSPBLK** Stratified random sampling with proportion data - inferences regarding the population proportion and total. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2c L4c | Usage: CALL SSPBLK (NBR,NH,IN,ALPHA,PH,STAT,IER) | On-line doc: CALL GAMSDOC SSPBLK (or @PRT IMSL*DOC.SSPBLK) | Access: LIB NBS*IMSL
- SSPCO** Computes factorization of real symmetric indefinite matrix stored in packed form and estimates its condition. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DSPCO. | Class(es): D2b1a | Usage: CALL SSPCO(AP,N,KPVT,RCOND,Z) | On-line doc: CALL GAMSDOC SSPCO (or @PRT CMLIB*DOC.SSPCO/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB
- SSPDI** Uses factorization of real symmetric indefinite matrix stored in packed form to compute its determinant and/or inverse. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DSPDI. | Class(es): D2b1a D3b1a | Usage: CALL SSPDI(AP,N,KPVT,DET,INERT,WORK,JOB) | On-line doc: CALL GAMSDOC SSPDI (or @PRT CMLIB*DOC.SSPDI/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SSPCO SSPFA
- SSPEV** Computes eigenvalues and, optionally eigenvectors of real symmetric matrix stored in packed form. | Portable single precision Fortran subprogram in LICEPACK sublibrary of CMLIB library. | Class(es): D4a1 | Usage: CALL SSPEV(A,N,E,V,LDV,WORK,JOB,INFO) | On-line doc: CALL GAMSDOC SSPEV (or @PRT CMLIB*DOC.SSPEV/LICEPACK) | Tests: CMLIB*TEST-SOURCE.SSPEV/LICEPACK | Access: LIB NBS*CMLIB
- SSPFA** Computes factorization of real symmetric indefinite matrix stored in packed form. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DSPFA. | Class(es): D2b1a | Usage: CALL SSPFA(AP,N,KPVT,INFO) | On-line doc: CALL GAMSDOC SSPFA (or @PRT CMLIB*DOC.SSPDI/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB
- SSPSL** Uses factorization of real symmetric indefinite matrix stored in packed form to solve systems. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DSPSL. | Class(es): D2b1a | Usage: CALL SSPSL(AP,N,KPVT,B) | On-line doc: CALL GAMSDOC SSPSL (or @PRT CMLIB*DOC.SSPSL/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SSPCO SSPFA
- SSRAND** Simple random sampling with continuous data - inferences regarding the population mean and total using ratio or regression estimation. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2c L4a1a | Usage: CALL SSRAND (Y,IY,IOPT,NBR,ALPHA,TEMP,XBAR,B,STAT,IER) | On-line doc: CALL GAMSDOC SSRAND (or @PRT IMSL*DOC.SSRAND) | Access: LIB NBS*IMSL
- SSSAND** Simple random sampling with continuous data - inferences regarding the population mean and total. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2c L4a1a | Usage: CALL SSSAND (Y,NBR,ALPHA,TEMP,STAT,IER) | On-line doc: CALL GAMSDOC SSSAND (or @PRT IMSL*DOC.SSSAND) | Access: LIB NBS*IMSL
- SSSBLK** Stratified random sampling with continuous data - inferences regarding the population mean and total. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2c L4a1a | Usage: CALL SSSBLK (Y,NBR,NH,IN,ALPHA,TEMP,HMUSIG,IH,STAT,IER) | On-line doc: CALL GAMSDOC SSSBLK (or @PRT IMSL*DOC.SSSBLK) | Access: LIB NBS*IMSL
- SSSCAN** Single stage cluster sampling with continuous data - inferences regarding the population mean and total. |

Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2c L4a1a | Usage: CALL SSSCAN (Y,IOPT,NBR,MC,IM,SIZE,TSIZE,ALPHA,TEMP,CMUSIG,IC, STAT,IER) | On-line doc: CALL GAMSDOC SSSCAN (or @PRT IMSL*DOC.SSSCAN) | Access: LIB NBS*IMSL

SSSEST Two-stage sampling with continuous data and equisized primary units - inferences regarding the population mean and total. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L2c L4a1a | Usage: CALL SSSEST (Y,NBR,ALPHA,TEMP,SMUSIG,IS,STAT,IER) | On-line doc: CALL GAMSDOC SSSEST (or @PRT IMSL*DOC.SSSEST) | Access: LIB NBS*IMSL

SSVDC Computes Singular Value Decomposition of real general matrix. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DSVDC. | Class(es): D6 | Usage: CALL SSVDC(X,LDX,N,P,S,E,U,LDU,V,LDV,WORK,JOB,INFO) | On-line doc: CALL GAMSDOC SSVDC (or @PRT CMLIB*DOC.SSVDC/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB

SSWAP Interchange vectors X and Y, both single precision. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Double precision version is DSWAP. | Class(es): D1a5 | Usage: CALL SSWAP(N,SX,INCX,SY,INCY) | On-line doc: CALL GAMSDOC SSWAP (or @PRT CMLIB*DOC.SSWAP/BLAS) | Tests: CMLIB*TEST-SOURCE.\$Q/BLAS | Access: LIB NBS*CMLIB

STATS Computes 53 descriptive statistics for a single random sample. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L1a1 | Usage: CALL STATS (Y, N, SCRAT, NS) | On-line doc: CALL GAMSDOC STATS (or @PRT STATLIB*DOC.STATS) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

STATSS Computes 53 descriptive statistics for a single random sample with individual weights assigned to each observation and computed results returned to the user. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L1a1 | Usage: CALL STATSS (Y, WT, N, SCRAT, NS, STAT, NPRT) | On-line doc: CALL GAMSDOC STATSS (or @PRT STATLIB*DOC.STATSS) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

STATSW Computes 53 descriptive statistics for a single random sample with individual weights assigned to each observation. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L1a1 | Usage: CALL STATSW (Y, WT, N, SCRAT, NS) | On-line doc: CALL GAMSDOC STATSW (or @PRT STATLIB*DOC.STATSW) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

STEM-and-LEAF Prints stem-and-leaf display(s), optionally with no trimming of outliers. | Command in MINITAB Proprietary interactive system. Class(es): L3d | Usage: STEM-and-leaf display of C,...,C ; subcommand NOTRIM.] | On-line doc: HELP STEM-AND-LEAF (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

STEPWISE Performs stepwise linear regression using forward selection, backward elimination, conventional stepwise, or user intervention. Options available through subcommands: F-to-enter and F-to-remove, force and remove sets of variables, print next "best" (by the F-statistic) K alternatives. | Command in MINITAB Proprietary interactive system. Class(es): L8a5 | Usage: STEPwise regression of y in C, predictors in C,...,C ; subcommands FENTER = K; FREMOVE = K; FORCE C,...,C; ENTER C,...,C; REMOVE C,...,C; BEST K; STEPS = K.] | On-line doc: HELP STEPWISE (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

STMOM3 Computes the sample standardized third central moment of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1c | Usage: CALL STMOM3(X,N,IWRITE,XSMOM3) | On-line doc: CALL GAMSDOC STMOM3 (or @PRT DATAPAC*DOC.STMOM3) | Access: LIB NBS*DATAPAC

STMOM4 Computes the sample standardized fourth central moment of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1c | Usage: CALL STMOM4(X,N,IWRITE,XSMOM4) | On-line doc: CALL GAMSDOC STMOM4 (or @PRT DATAPAC*DOC.STMOM4) | Access: LIB NBS*DATAPAC

STRCO Estimates the condition of real triangular matrix. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DTRCO. | Class(es): D2a3 | Usage: CALL STRCO(T,LDT,N,RCOND,Z,JOB) | On-line doc: CALL GAMSDOC STRCO (or @PRT CMLIB*DOC.STRCO/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB

STRDI Computes determinant and/or inverse of real triangular matrix. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DTRDI. | Class(es): D2a3 D3a3 | Usage: CALL STRDI(T,LDT,N,DET,JOB,INFO) | On-line doc: CALL GAMSDOC STRDI (or @PRT CMLIB*DOC.STRDI/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB

STRSL Solves systems with real triangular matrix. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DTRSL. | Class(es): D2a3 | Usage: CALL STRSL(T,LDT,N,B,JOB,INFO) | On-line doc: CALL GAMSDOC STRSL (or @PRT CMLIB*DOC.STRSL/LINPACKS) | Tests: CMLIB*TEST-SOURCE.\$Q/LINPACKS, CMLIB*TEST-SOURCE.\$F/LINPACKS | Access: LIB NBS*CMLIB

SUBSE1 Carry over into Y all observations of vector X for which the corresponding elements in vector D are in the interval [DMIN,DMAX]. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L2d | Usage: CALL SUBSE1(X,N,D,DMIN,DMAX,Y,NY) | On-line doc: CALL GAMSDOC SUBSE1 (or @PRT DATAPAC*DOC.SUBSE1) | Access: LIB NBS*DATAPAC

- SUBSE2** Carry over into Y all observations of vector X for which the corresponding elements in vector D1 are in the inclusive interval [D1MIN,D1MAX] and also for which the corresponding elements in D2 are in the interval [D2MIN,D2MAX]. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L2d | Usage: CALL SUBSE2(X,N,D1,D1MIN,D1MAX,D2,D2MIN,D2MAX,Y,NY) | On-line doc: CALL GAMSDOC SUBSE2 (or @PRT DATAPAC*DOC.SUBSE2) | Access: LIB NBS*DATAPAC
- SUBSET** Retain all observations in vector X for which the corresponding elements in vector D are in the interval [DMIN,DMAX]. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L2d | Usage: CALL SUBSET (X,N,D,DMIN,DMAX,NEWN) | On-line doc: CALL GAMSDOC SUBSET (or @PRT DATAPAC*DOC.SUBSET) | Access: LIB NBS*DATAPAC
- SUDS** Solves underdetermined systems of linear equations. For full rank matrices the minimum norm solution is returned, as well as an orthonormal basis for the null space of the matrix. If the system of equations is inconsistent only the least squares solution of minimal length is computed. | Portable single precision Fortran subprogram in SUDSSODS sublibrary of CMLIB library. | Class(es): D9 | Usage: CALL SUDS(A,X,B,NEQ,NUK,NRDA,IFLAG,MLSO,WORK,IWORK) | On-line doc: CALL GAMSDOC SUDS (or @PRT CMLIB*DOC.SUDS/SUDSSODS) | Tests: CMLIB*TEST-SOURCE.SUDS/SUDSSODS | Access: LIB NBS*CMLIB
- SUMSL** Minimizes a general unconstrained objective function using analytic gradient and a Hessian approximation from a secant update. | Portable single precision Fortran subprogram in NL2SN sublibrary of CMLIB library. Double precision version is DSUMSL. | Class(es): G1b1b | Usage: CALL SUMSL(N,D,X,CALCF,GALCG,IV,LIV,LV,V,UIPARM,URPARM,UFPARM) | On-line doc: CALL GAMSDOC SUMSL (or @PRT CMLIB*DOC.SUMSL/NL2SN) | Tests: CMLIB*TEST-SOURCE.\$F2/NL2SN, CMLIB*TEST-SOURCE.\$Q2/NL2SN | Access: LIB NBS*CMLIB
- SVD** Compute Singular Value Decomposition of arbitrary real rectangular matrix. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D6 | Usage: CALL SVD(NM,M,N,A,W,MATU,U,MATV,V,IERR,RV1) | On-line doc: CALL GAMSDOC SVD (or @PRT CMLIB*DOC.SVD/EISPACK) | Access: LIB NBS*CMLIB

T

- TABLE** Produces and prints one-way, two-way, and multi-way tables of counts with 20 optional subcommands for summarizing (e.g., cell mean, standard deviation), marginals, performing chi-square tests for each 2-way table, handling missing values, and selecting forms of input and output. | Command in MINITAB Proprietary interactive system. Class(es): L1e1 L1e2 L9 | Usage: TABLE the data classified by C,...; | subcommands MEANS for C, ...,C; MEDIANs ...; SUMS ...; MINIMUMS ...; MAXIMUMS ...; STDEV ...; STATS ...; DATA ...; NONMISSING ...; PROPORTION ...; COUNTS; ROWPERCENTS; COLPERCENTS; TOTPERCENTS; CHISQUARE ...; NOALL; ALL ...; MISSING ...; FREQUENCIES ...; LAYOUT ... | On-line doc: HELP TABLE (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- TAIL** Performs a symmetric distribution tail length analysis on the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L4a1a | Usage: CALL TAIL(X,N) | On-line doc: CALL GAMSDOC TAIL (or @PRT DATAPAC*DOC.TAIL) | Access: LIB NBS*DATAPAC
- TAN** Computes the elementary tangent function. If your Fortran library includes this function, use that instead. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DTAN. | Class(es): C4a | Usage: X = TAN (X) | On-line doc: CALL GAMSDOC TAN (or @PRT PORT*DOC.TAN) | Access: LIB NBS*PORT
- TANH** Computes hyperbolic tangent, $\tanh(x)$. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DTANH. | Class(es): C4c | Usage: X = TANH (X) | On-line doc: CALL GAMSDOC TANH (or @PRT PORT*DOC.TANH) | Access: LIB NBS*PORT
- TCDF** Computes the cumulative distribution function value for Student's t distribution with degrees of freedom parameter NU. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1t | Usage: CALL TCDF(X,NU,CDF) | On-line doc: CALL GAMSDOC TCDF (or @PRT DATAPAC*DOC.TCDF) | Access: LIB NBS*DATAPAC
- TCHBP** Evaluates a polynomial expressed as a sum of Chebyshev polynomials. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DTCHBP. | Class(es): C3a2 | Usage: X = TCHBP (N, ALPHA, X, X0, X1) | On-line doc: CALL GAMSDOC TCHBP (or @PRT PORT*DOC.TCHBP) | Access: LIB NBS*PORT
- TDRV** Solves sparse systems of linear algebraic equations by Gaussian elimination without pivoting. | Portable single precision Fortran subprogram in YSMP sublibrary of CMLIB library. | Class(es): D2a4 | Usage: CALL TDRV(N,R,IC,1A,JA,A,B,Z,NSP,ISF,RSP,ESP,FLAG) | On-line doc: CALL GAMSDOC TDRV (or @PRT CMLIB*DOC.TDRV/YSMP) | Tests: CMLIB*TEST-SOURCE.\$Q2/YSMP | Access: LIB NBS*CMLIB
- TIME** Performs a time series analysis on the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L10c | Usage: CALL TIME(X,N) | On-line doc: CALL GAMSDOC TIME (or @PRT DATAPAC*DOC.TIME) | Access: LIB NBS*DATAPAC
- TINTERVAL** Calculates a t-confidence interval with specified percent confidence. | Command in MINITAB Proprietary interactive system. Class(es): L4a1a14 L4a2 | Usage: TINterval [with K percent confidence] for data in column C | On-line doc: HELP TINTERVAL (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- TINVIT** Eigenvectors of symmetric tridiagonal matrix corresponding to some specified eigenvalues, using inverse iteration. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c3 | Usage: CALL TINVIT(NM,N,D,E,E2,M,W,IND,Z,IERR,RV1,RV2,RV3,RV4,RV6) | On-line doc: CALL GAMSDOC TINVIT (or @PRT CMLIB*DOC.TINVIT/EISPACK) | Access: LIB NBS*CMLIB
- TOL** Computes normal and distribution-free tolerance limits for the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L4a1f | Usage: CALL TOL(X,N) | On-line doc: CALL GAMSDOC TOL (or @PRT DATAPAC*DOC.TOL) | Access: LIB NBS*DATAPAC
- TPLOT** Prints pseudo three-dimensional plot of y versus x versus z, with symbols indicating the values of z, and with optional scale specification. | Command in MINITAB Proprietary interactive system. Class(es): L3c2 Q1 | Usage: TPLOT y in C [from K to K] vs x in C [from K to K] vs z in C | On-line doc: HELP TPLOT (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- TPLT** Generates a Student's t probability plot with degrees of freedom parameter NU. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c4t | Usage: CALL TPLT(X,N,NU) | On-line doc: CALL GAMSDOC TPLT (or @PRT DATAPAC*DOC.TPLT) | Access: LIB NBS*DATAPAC
- TPPF** Computes the percent point function value for the Student's t distribution with degrees of freedom parameter NU. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2t | Usage: CALL TPPF(P,NU,PPF) | On-line doc: CALL GAMSDOC TPPF (or @PRT DATAPAC*DOC.TPPF) | Access: LIB NBS*DATAPAC
- TQL1** Compute eigenvalue of symmetric tridiagonal matrix by QL method. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 D4c2a | Usage: CALL TQL1(N,D,E,IERR) | On-line doc: CALL GAMSDOC TQL1 (or @PRT CMLIB*DOC.TQL1/EISPACK) | Access: LIB NBS*CMLIB
- TQL2** Compute eigenvalues and eigenvectors of symmetric tridiagonal matrix. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 D4c2a | Usage: CALL TQL2(NM,N,D,E,Z,IERR) | On-line doc: CALL GAMSDOC TQL2

(or @PRT CMLIB*DOC.TQL2/EISPACK) | Access: LIB NBS*CMLIB

TQLRAT Computes eigenvalues of symmetric tridiagonal matrix using a rational variant of the QL method. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 D4c2a | Usage: CALL TQLRAT(N,D,E2,IERR) | On-line doc: CALL GAMSDOC TQLRAT (or @PRT CMLIB*DOC.TQLRAT/EISPACK) | Access: LIB NBS*CMLIB

TRAN Generates a random sample of size N from the Student's t distribution with degrees of freedom parameter NU. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a20 | Usage: CALL TRAN(N,NU,ISTART,X) | On-line doc: CALL GAMSDOC TRAN (or @PRT DATAPAC*DOC.TRAN) | Access: LIB NBS*DATAPAC

TRBAK1 Forms the eigenvectors of real symmetric matrix from eigenvectors of symmetric tridiagonal matrix formed by TRED1. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL TRBAK1(NM,N,A,E,M,Z) | On-line doc: CALL GAMSDOC TRBAK1 (or @PRT CMLIB*DOC.TRBAK1/EISPACK) | Access: LIB NBS*CMLIB | See also: TRED1

TRBAK3 Forms eigenvectors of real symmetric matrix from the eigenvectors of symmetric tridiagonal matrix formed by TRED3. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL TRBAK3(NM,N,NV,A,M,Z) | On-line doc: CALL GAMSDOC TRBAK3 (or @PRT CMLIB*DOC.TRBAK3/EISPACK) | Access: LIB NBS*CMLIB | See also: TRED3

TRED1 Reduce real symmetric matrix to symmetric tridiagonal matrix using orthogonal similarity transformations. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1b1 | Usage: CALL TRED1(NM,N,A,D,E,E2) | On-line doc: CALL GAMSDOC TRED1 (or @PRT CMLIB*DOC.TRED1/EISPACK) | Access: LIB NBS*CMLIB

TRED2 Reduce real symmetric matrix to symmetric tridiagonal matrix using and accumulating orthogonal transformations. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1b1 | Usage: CALL TRED2(NM,N,A,D,E,Z) | On-line doc: CALL GAMSDOC TRED2 (or @PRT CMLIB*DOC.TRED2/EISPACK) | Access: LIB NBS*CMLIB | See also: TQL2,IMTQL2

TRED3 Reduce real symmetric matrix stored in packed form to symmetric tridiagonal matrix using orthogonal transformations. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1b1 | Usage: CALL TRED3(N,NV,A,D,E,E2) | On-line doc: CALL GAMSDOC TRED3 (or @PRT CMLIB*DOC.TRED3/EISPACK) | Access: LIB NBS*CMLIB

TRIDIB Computes eigenvalues of symmetric tridiagonal matrix in given interval using Sturm sequencing. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 D4c2a | Usage: CALL TRIDIB(N,EPS1,D,E,E2,LB,UB,M11,M,W,IND,IERR,RV4,RV5) | On-line doc: CALL GAMSDOC TRIDIB (or @PRT CMLIB*DOC.TRIDIB/EISPACK) | Access: LIB NBS*CMLIB

TRIGP Evaluates a trigonometric polynomial with given coefficients. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DTRIGP. | Class(es): C3a1 | Usage: X = TRIGP (N, ALPHA, BETA, THETA) | On-line doc: CALL GAMSDOC TRIGP (or @PRT PORT*DOC.TRIGP) | Access: LIB NBS*PORT

TRIM Computes the sample trimmed mean of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1a | Usage: CALL TRIM(X,N,P1,P2,IWRITE,XTRIM) | On-line doc: CALL GAMSDOC TRIM (or @PRT DATAPAC*DOC.TRIM) | Access: LIB NBS*DATAPAC

TSDIFF Performs a user-controlled differencing operation on a series. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L10b | Usage: CALL TSDIFF (Y, N, NDIFAC, ND, IOD, DIFF, NDIFF) | On-line doc: CALL GAMSDOC TSDIFF (or @PRT STATLIB*DOC.TSDIFF) | Tests: STATLIB*TEST.DEMO3 | Access: LIB NBS*STATLIB

TSPLLOT Prints a scatter diagram of a time series, optionally using symbols modulo the period. Handles missing values. | Command in MINITAB Proprietary interactive system. Class(es): L3c5 Q1 | Usage: TSPLot [with period K [starting at K]] for data in C | On-line doc: HELP TSPLLOT (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

TSTURM Computes eigenvalues of symmetric tridiagonal matrix in given interval and eigenvectors by Sturm sequencing. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 D4c2a | Usage: CALL TSTURM(NM,N,EPS1,D,E,E2,LB,UB,MN,M,W,Z,IERR,RV1,RV2,RV3, RV4,RV5,RV6) | On-line doc: CALL GAMSDOC TSTURM (or @PRT CMLIB*DOC.TSTURM/EISPACK) | Access: LIB NBS*CMLIB

TTEST Performs one- or two-sided t-tests. | Command in MINITAB Proprietary interactive system. Class(es): L4a1a14 L4a2 | Usage: TTEST [of mu = K] on data in C,...,C [; ALTERNATIVE = K.] | On-line doc: HELP TTEST (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

TWOSAMPLE Performs a one- or two-sided two-sample t-test of equality of two population means with variance estimated either from each sample or the pooled sample. | Command in MINITAB Proprietary interactive system. Class(es): L4b1a14 L4b2 | Usage: TWOSample t [K percent confidence] for data in C and C [; subcommands ALTERNATIVE = K; POOLED.] | On-line doc: HELP TWOSAMPLE (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

TWOT Performs a one- or two-sided two-sample t-test of equality of two population means with variance estimated either from each sample

or the pooled sample. | Command in MINITAB Proprietary interactive system. Class(es): L4b1a14 L4b2 | Usage: TWOT [K percent confidence] for data in C, groups in C [; subcommands ALTERNATIVE = K; POOLED.] | On-line doc: HELP TWOT (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

TWOWAYAOV Performs two-way analysis of variance for balanced data (equal number of observations, one or more, in each cell) and prints standard results. Options: fit additive model, save results. | Command in MINITAB Proprietary interactive system. Class(es): L7a2a1a | Usage: TWOWayaov for data in C, subscripts in C, C [store residuals in C [fits in C]] [; subcommand ADDITIVE.] | On-line doc: HELP TWOWAYAOV (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

U

- UERSSET** Set message level for IMSL routine UERTST. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): R3a | Usage: CALL UERSSET (LEVEL,LEVOLD) | On-line doc: CALL GAMSDOC UERSSET (or @PRT IMSL*DOC.UERSSET) | Access: LIB NBS*IMSL
- UERTST** Print a message reflecting an error condition. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): R3c | Usage: CALL UERTST (IER,NAME) | On-line doc: CALL GAMSDOC UERTST (or @PRT IMSL*DOC.UERTST) | Access: LIB NBS*IMSL
- UGETIO** To retrieve current values and to set new values for input and output unit identifiers. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): R1 R3b | Usage: CALL UGETIO (IOPT,NIN,NOUT) | On-line doc: CALL GAMSDOC UGETIO (or @PRT IMSL*DOC.UGETIO) | Access: LIB NBS*IMSL
- UHELP** Display methods of obtaining info on IMSL conventions regarding various subjects provide means for individual sites to supply users with site specific info. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): R4 | Usage: CALL UHELP | On-line doc: CALL GAMSDOC UHELP (or @PRT IMSL*DOC.UHELP) | Access: LIB NBS*IMSL
- UHELP1** Write information regarding IMSL conventions and notation to an output file. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): R4 | Usage: CALL UHELP1 | On-line doc: CALL GAMSDOC UHELP1 (or @PRT IMSL*DOC.UHELP1) | Access: LIB NBS*IMSL
- UHELP2** Write information regarding IMSL input and output conventions. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): R4 | Usage: CALL UHELP2 | On-line doc: CALL GAMSDOC UHELP2 (or @PRT IMSL*DOC.UHELP2) | Access: LIB NBS*IMSL
- UHELP3** Write information regarding IMSL error detecting facilities. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): R4 | Usage: CALL UHELP3 | On-line doc: CALL GAMSDOC UHELP3 (or @PRT IMSL*DOC.UHELP3) | Access: LIB NBS*IMSL
- UHELP4** Write information regarding matrix/vector storage modes used in IMSL subroutines. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): R4 | Usage: CALL UHELP4 | On-line doc: CALL GAMSDOC UHELP4 (or @PRT IMSL*DOC.UHELP4) | Access: LIB NBS*IMSL
- ULSIA** Finds the minimal length solution of the undetermined system of equations $AX=B$ where A is an m by n matrix with m.le.n Flexible version of SGLSS. | Portable single precision Fortran subprogram in SGLSS sublibrary of CMLIB library. | Class(es): D9 | Usage: CALL ULSIA(A,MDA,M,N,B,MDB,NB,RE,AE,KEY,MODE,NP, KRANK,KSURE, RNORM,W,LW,IWORK,LIW,INFO) | On-line doc: CALL GAMSDOC ULSIA (or @PRT CMLIB*DOC.ULSIA/SGLSS) | Access: LIB NBS*CMLIB
- UMB** Given interval endpoints, this generates a uniform mesh, with needed multiplicities for B-spline use. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DUMB. | Class(es): E3 K6 | Usage: CALL UMB (A,B,NAB,K,X,NX) | On-line doc: CALL GAMSDOC UMB (or @PRT PORT*DOC.UMB) | Access: LIB NBS*PORT
- UMD** Given interval endpoints, this generates a uniform mesh of distinct points. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DUMD. | Class(es): E3 K6 | Usage: CALL UMD (A,B,NAB,X) | On-line doc: CALL GAMSDOC UMD (or @PRT PORT*DOC.UMD) | Access: LIB NBS*PORT
- UMKFL** Decomposes a non-zero floating point number into a mantissa and an exponent. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DUMKFL. | Class(es): A6c | Usage: CALL UMKFL (F, E, M) | On-line doc: CALL GAMSDOC UMKFL (or @PRT PORT*DOC.UMKFL) | Access: LIB NBS*PORT
- UNI** Generates uniformly distributed random numbers on the interval [0,1). UNI's main advantages are a long cycle and a high degree of reproducibility on other machines (it runs on any machine with at least 16 bit integer arithmetic). | Portable single precision Fortran subprogram in RV sublibrary of CMLIB library. | Class(es): L6a21 | Usage: S=UNI(JD) | On-line doc: CALL GAMSDOC UNI (or @PRT CMLIB*DOC.UNI/RV) | Tests: CMLIB*TEST-SOURCE.UNI/UNI | Access: LIB NBS*CMLIB
- UNI** Returns a single real random variate from the uniform [0,1) distribution. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): L6a21 | Usage: X = UNI (K) | On-line doc: CALL GAMSDOC UNI (or @PRT PORT*DOC.UNI) | Access: LIB NBS*PORT | See also: RANBYT,RANSET
- UNICDF** Computes the cumulative distribution function value for the uniform (rectangular) distribution on the unit interval (0,1). | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1u | Usage: CALL UNICDF(X,CDF) | On-line doc: CALL GAMSDOC UNICDF (or @PRT DATAPAC*DOC.UNICDF) | Access: LIB NBS*DAPAC
- UNIPDF** Computes the probability density function value for the uniform (rectangular) distribution on the unit interval (0,1). | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1u | Usage: CALL UNIPDF(X,PDF) | On-line doc: CALL GAMSDOC UNIPDF (or @PRT DATAPAC*DOC.UNIPDF) | Access: LIB NBS*DAPAC
- UNIPLT** Generates a uniform probability plot on the unit interval (0,1) with mean = 0.5 and standard deviation = sqrt(1/12). | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c4u | Usage: CALL UNIPLT(X,N) | On-line doc: CALL GAMSDOC UNIPLT (or @PRT DATAPAC*DOC.UNIPLT) | Access: LIB NBS*DAPAC

- UNIPPF** Computes the percent point function value for the uniform (rectangular) distribution on the unit interval (0,1). | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2u | Usage: CALL UNIPPF(P,PPF) | On-line doc: CALL GAMSDOC UNIPPF (or @PRT DATAPAC*DOC.UNIPPF) | Access: LIB NBS*DAPAC
- UNIRAN** Generates a random sample of size N from the uniform (rectangular) distribution on the unit interval (0,1). | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a21 | Usage: CALL UNIRAN(N,ISTART,X) | On-line doc: CALL GAMSDOC UNIRAN (or @PRT DATAPAC*DOC.UNIRAN) | Access: LIB NBS*DAPAC
- UNISF** Computes the sparsity function value for the uniform (rectangular) distribution on the unit interval (0,1). | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2u | Usage: CALL UNISF(P,SF) | On-line doc: CALL GAMSDOC UNISF (or @PRT DATAPAC*DOC.UNISF) | Access: LIB NBS*DAPAC
- URANDOM** Generates K pseudo-random numbers from the uniform (0,1) distribution. | Command in MINITAB Proprietary interactive system. Class(es): L6a21 | Usage: URANDOM K observations, put into C | On-line doc: HELP URANDOM (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- USBOX** Print a boxplot (k samples). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L3d | Usage: CALL USBOX (X,K,NI,MAXL,IER) | On-line doc: CALL GAMSDOC USBOX (or @PRT IMSL*DOC.USBOX) | Access: LIB NBS*IMSL
- USCWV** Print a complex vector. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N1 | Usage: CALL USCWV(ITITLE,NC,A,M,INC,IOPT) | On-line doc: CALL GAMSDOC USCWV (or @PRT IMSL*DOC.USCWV) | Access: LIB NBS*IMSL
- USHHST** Print a horizontal histogram. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L3a Q1 | Usage: CALL USHHST (T, N, IOPT, IER) | On-line doc: CALL GAMSDOC USHHST (or @PRT IMSL*DOC.USHHST) | Access: LIB NBS*IMSL
- USHST** Print a vertical histogram. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L3a Q1 | Usage: CALL USHST (T, N, ISP, IER) | On-line doc: CALL GAMSDOC USHST (or @PRT IMSL*DOC.USHST) | Access: LIB NBS*IMSL
- USHST2** Print a vertical histogram, plotting two frequencies with one bar of the histogram. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L3a | Usage: CALL USHST2 (T, U, N, ISP, IER) | On-line doc: CALL GAMSDOC USHST2 (or @PRT IMSL*DOC.USHST2) | Access: LIB NBS*IMSL
- USLEAP** Print results of the best-regressions analysis performed by IMSL routine RLEAP. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8h | Usage: CALL USLEAP (IJOB,KZ,IXS,STAT,IXV,NVAR,IXB,BEST,IB) | On-line doc: CALL GAMSDOC USLEAP (or @PRT IMSL*DOC.USLEAP) | Access: LIB NBS*IMSL | See also: RLEAP
- USNMNMX** Determination of the minimum and maximum values of a vector. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1a2 | Usage: CALL USNMNMX (X,N,INC,XMIN,XMAX) | On-line doc: CALL GAMSDOC USNMNMX (or @PRT IMSL*DOC.USNMNMX) | Access: LIB NBS*IMSL
- USPC** Print a sample pdf, a theoretical pdf and confidence band information; plot these on option. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1e L3b | Usage: CALL USPC (PDF,X,N,N12,N05,IP,IC,W) | On-line doc: CALL GAMSDOC USPC (or @PRT IMSL*DOC.USPC) | Access: LIB NBS*IMSL
- USPDF** Plot of two sample probability distribution functions against their spectra. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L3b | Usage: CALL USPDF (X,N,M,W,IW,IR) | On-line doc: CALL GAMSDOC USPDF (or @PRT IMSL*DOC.USPDF) | Access: LIB NBS*IMSL
- USPLO** Printer plot of up to ten functions. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L3b Q1 | Usage: CALL USPLO (X, Y, IY, N, M, INC, ITITLE, NTITLE, IXLABL, NXLABL, IYLABL, NYLABL, RANGE, ICHAR, IOPT, IER) | On-line doc: CALL GAMSDOC USPLO (or @PRT IMSL*DOC.USPLO) | Access: LIB NBS*IMSL
- USPLOD** Printer plot of up to ten functions. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): L3b Q1 | Usage: CALL USPLOD (X, Y, IY, N, M, INC, ITITLE, NTITLE, IXLABL, NXLABL, IYLABL, NYLABL, RANGE, ICHAR, IOPT, IER) | On-line doc: CALL GAMSDOC USPLOD (or @PRT IMSL*DOC.USPLOD) | Access: LIB NBS*IMSL
- USPRP** Probability plot. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L3c4 | Usage: CALL USPRP (X, N, N1, N2, IDIST, IOPT, WK, IER) | On-line doc: CALL GAMSDOC USPRP (or @PRT IMSL*DOC.USPRP) | Access: LIB NBS*IMSL
- USSLF** Print a stem-and-leaf display. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L3d | Usage: CALL USSLF (X,N,IUNIT,MAXL) | On-line doc: CALL GAMSDOC USSLF (or @PRT IMSL*DOC.USSLF) | Access: LIB NBS*IMSL
- USTREE** Print a binary tree (which may represent the output of a clustering algorithm in chapter O). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L14c N4 Q1 | Usage: CALL USTREE (ND,ICLSON,ICRSON,CLEVEL,IND,XSIM,IOUT,CLVLSK, NCLRST,LEFTRT,STARST,I) | On-line doc: CALL GAMSDOC USTREE (or @PRT IMSL*DOC.USTREE) | Access: LIB NBS*IMSL
- USWBM** Print a matrix stored in band storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N1 | Usage: CALL USEBM (ITITLE,NC,A,IA,M,NUC,NLC,WK,IOPT) | On-line doc: CALL GAMSDOC USWBM (or @PRT IMSL*DOC.USWBM)

| Access: LIB NBS*IMSL

USWBS Print a matrix stored in band symmetric storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N1 | Usage: CALL USWBS (ITITLE,NC,A,IA,M,NLC,WK,IOPT) | On-line doc: CALL GAMSDOC USWBS (or @PRT IMSL*DOC.USWBS) | Access: LIB NBS*IMSL

USWCH Print a complex matrix stored in Hermitian storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N1 | Usage: CALL USWCH(ITITLE,NC,A,M,IOPT) | On-line doc: CALL GAMSDOC USWCH (or @PRT IMSL*DOC.USWCH) | Access: LIB NBS*IMSL

USWCM Print a complex matrix stored in full storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N1 | Usage: CALL USCWM(ITITLE,NC,A,IA,N,M,IOPT) | On-line doc: CALL GAMSDOC USWCM (or @PRT IMSL*DOC.USWCM) | Access: LIB NBS*IMSL

USWFM Print a matrix stored in full storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N1 | Usage: CALL USWFM (ITITLE,NC,A,IA,N,M,IOPT) | On-line doc: CALL GAMSDOC USWFM (or @PRT IMSL*DOC.USWFM) | Access: LIB NBS*IMSL

USWFV Print a vector. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N1 | Usage: CALL USWFV (ITITLE,NC,A,M,INC,IOPT) | On-line doc: CALL GAMSDOC USWFV (or @PRT IMSL*DOC.USWFV) | Access: LIB NBS*IMSL

USWSM Print a matrix stored in symmetric storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N1 | Usage: CALL USWSM (ITITLE,NC,A,M,IOPT) | On-line doc: CALL GAMSDOC USWSM (or @PRT IMSL*DOC.USWSM) | Access: LIB NBS*IMSL

V

- VABMXF** Maximum absolute value of the elements of a vector or a subset of the elements of a vector. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1a3c | Usage: CALL VABMXF (V,L,INC,J,VMAX) | On-line doc: CALL GAMSDOC VABMXF (or @PRT IMSL*DOC.VABMXF) | Access: LIB NBS*IMSL
- VABMXS** Maximum absolute value of the elements of a row or column of a matrix stored in symmetric storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1a3c | Usage: CALL VABMXS (V,L,IRO,J,VMAX) | On-line doc: CALL GAMSDOC VABMXS (or @PRT IMSL*DOC.VABMXS) | Access: LIB NBS*IMSL
- VABSMF** Sum of the absolute values of the elements of a vector or a subset of a vector. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1a3a | Usage: CALL VABSMF (V,L,INC,VSUM) | On-line doc: CALL GAMSDOC VABSMF (or @PRT IMSL*DOC.VABSMF) | Access: LIB NBS*IMSL
- VABSMS** Sum of the absolute values of the elements of a row (or column) of a matrix stored in symmetric storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1a3a | Usage: CALL VABSMS (V,L,IRO,VSUM) | On-line doc: CALL GAMSDOC VABSMS (or @PRT IMSL*DOC.VABSMS) | Access: LIB NBS*IMSL
- VAR** Computes the sample variance (with denominator N-1) of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1b | Usage: CALL VAR(X,N,IWRITE,XVAR) | On-line doc: CALL GAMSDOC VAR (or @PRT DATAPAC*DOC.VAR) | Access: LIB NBS*DAPAC
- VBOD** Converts a mantissa and exponent into a base 10 floating point number. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DVBTOD. | Class(es): A6b | Usage: CALL VBOD (E, M, E10, M10) | On-line doc: CALL GAMSDOC VBOD (or @PRT PORT*DOC.VBTOD) | Access: LIB NBS*PORT
- VCONVO** Vector convolution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1a10 J2 | Usage: CALL VCONVO (A,B,LA,LB,IWK) | On-line doc: CALL GAMSDOC VCONVO (or @PRT IMSL*DOC.VCONVO) | Access: LIB NBS*IMSL
- VCVTBF** Storage mode conversion of matrices (band to full storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b9 | Usage: CALL VCVTBF (A,N,NUC,NLC,IA,B,IB) | On-line doc: CALL GAMSDOC VCVTBF (or @PRT IMSL*DOC.VCVTBF) | Access: LIB NBS*IMSL
- VCVTCH** Storage mode conversion of matrices (full complex to Hermitian). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b9 | Usage: CALL VCVTCH (A,N,IA,H) | On-line doc: CALL GAMSDOC VCVTCH (or @PRT IMSL*DOC.VCVTCH) | Access: LIB NBS*IMSL
- VCVTFB** Storage mode conversion of matrices (full to band storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b9 | Usage: CALL VCVTFB (A,N,NUC,NLC,IA,B,IB) | On-line doc: CALL GAMSDOC VCVTFB (or @PRT IMSL*DOC.VCVTFB) | Access: LIB NBS*IMSL
- VCVTFQ** Storage mode conversion (full to band symmetric storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b9 | Usage: CALL VCVTFQ (A,N,NC,IA,AA,IAA) | On-line doc: CALL GAMSDOC VCVTFQ (or @PRT IMSL*DOC.VCVTFQ) | Access: LIB NBS*IMSL
- VCVTFS** Storage mode conversion of matrices (full to symmetric). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b9 | Usage: CALL VCVTFS (A,N,IA,B) | On-line doc: CALL GAMSDOC VCVTFS (or @PRT IMSL*DOC.VCVTFS) | Access: LIB NBS*IMSL
- VCVTHC** Storage mode conversion of matrices (Hermitian to full complex). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b9 | Usage: CALL VCVTHC (H,N,B,IB) | On-line doc: CALL GAMSDOC VCVTHC (or @PRT IMSL*DOC.VCVTHC) | Access: LIB NBS*IMSL
- VCVTQF** Storage mode conversion (band symmetric to full storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b9 | Usage: CALL VCVTQF (A,N,NC,IA,B,IB) | On-line doc: CALL GAMSDOC VCVTQF (or @PRT IMSL*DOC.VCVTQF) | Access: LIB NBS*IMSL
- VCVTQS** Storage mode conversion (band symmetric to symmetric storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b9 | Usage: CALL VCVTQS (A,N,NC,IA,B) | On-line doc: CALL GAMSDOC VCVTQS (or @PRT IMSL*DOC.VCVTQS) | Access: LIB NBS*IMSL
- VCVTSF** Storage mode conversion of matrices (symmetric to full). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b9 | Usage: CALL VCVTSF (A,N,B,IB) | On-line doc: CALL GAMSDOC VCVTSF (or @PRT IMSL*DOC.VCVTSF) | Access: LIB NBS*IMSL
- VCVTSQ** Storage mode conversion (symmetric to band symmetric storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b9 | Usage: CALL VCVTSQ (A,N,NC,B,IB) | On-line doc: CALL GAMSDOC VCVTSQ (or @PRT IMSL*DOC.VCVTSQ) | Access: LIB NBS*IMSL
- VDCPS** Decompose an integer into its prime factors. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): B A1 | Usage: CALL VDCPS (N,NPF,IPF,LEX,IPWR) | On-line doc: CALL GAMSDOC VDCPS (or @PRT IMSL*DOC.VDCPS) | Access:

LIB NBS*IMSL

- VDTOB** Converts a base-10 mantissa and exponent of a floating point number into a machine-base representation. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DVBTOb. | Class(es): A6c | Usage: CALL VDTOb (E10, M10, E, M) | On-line doc: CALL GAMSDOC VDTOb (or @PRT PORT*DOC.VDTOb) | Access: LIB NBS*PORT
- VHS12** Real Householder transformation computation and applications. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1a9 D1b11 | Usage: CALL VHS12 (MODE,LP,L1,M,U,INCU,UP,C,INCC,ICV,NCV) | On-line doc: CALL GAMSDOC VHS12 (or @PRT IMSL*DOC.VHS12) | Access: LIB NBS*IMSL
- VHSH2C** Complex Householder transformation to zero a single element of a matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b11 | Usage: CALL VHSH2C (AJR,AJI,AJP1R,AJP1I,C,SR,SI) | On-line doc: CALL GAMSDOC VHSH2C (or @PRT IMSL*DOC.VHSH2C) | Access: LIB NBS*IMSL
- VHSH2R** Real Householder transformation to zero a single element of a matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b11 | Usage: CALL VHSH2R (AJ,AJP1,UJ,UJP1,VJ,VJP1) | On-line doc: CALL GAMSDOC VHSH2R (or @PRT IMSL*DOC.VHSH2R) | Access: LIB NBS*IMSL
- VHSH3R** Real Householder transformation to zero two elements of a matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b11 | Usage: CALL VHSH3R (AJ,AJP1,AJP2,UJ,UJP1,UJP2,VJ,VJP1,VJP2) | On-line doc: CALL GAMSDOC VHSH3R (or @PRT IMSL*DOC.VHSH3R) | Access: LIB NBS*IMSL
- VIPRFF** Vector inner product of two vectors or subsets of two vectors. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1a4 | Usage: CALL VIPRFF (X,Y,L,IX,IY,XYIP) | On-line doc: CALL GAMSDOC VIPRFF (or @PRT IMSL*DOC.VIPRFF) | Access: LIB NBS*IMSL
- VIPRSS** Vector inner product of two vectors each of which is part of some matrix stored in symmetric mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1a4 | Usage: CALL VIPRSS (X,Y,L,IX,IY,XYIP) | On-line doc: CALL GAMSDOC VIPRSS (or @PRT IMSL*DOC.VIPRSS) | Access: LIB NBS*IMSL
- VMULBB** Matrix multiplication (band storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULBB (A,IA,B,IB,N,C,IC) | On-line doc: CALL GAMSDOC VMULBB (or @PRT IMSL*DOC.VMULBB) | Access: LIB NBS*IMSL
- VMULBF** Matrix multiplication (band by full matrices). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULBF (A,IA,B,IB,N,C,IC) | On-line doc: CALL GAMSDOC VMULBF (or @PRT IMSL*DOC.VMULBF) | Access: LIB NBS*IMSL
- VMULBS** Matrix multiplication (band by symmetric matrices). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULBS (A,IA,B,N,C,IC) | On-line doc: CALL GAMSDOC VMULBS (or @PRT IMSL*DOC.VMULBS) | Access: LIB NBS*IMSL
- VMULFB** Matrix multiplication (full by band matrices). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULFB (A,IA,B,IB,N,C,IC) | On-line doc: CALL GAMSDOC VMULFB (or @PRT IMSL*DOC.VMULFB) | Access: LIB NBS*IMSL
- VMULFF** Matrix multiplication (full storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULFF (A,B,L,M,N,IA,IB,C,IC,IER) | On-line doc: CALL GAMSDOC VMULFF (or @PRT IMSL*DOC.VMULFF) | Access: LIB NBS*IMSL
- VMULFM** Matrix multiplication of the transpose of matrix a by matrix b (full storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULFM (A,B,L,M,N,IA,IB,C,IC,IER) | On-line doc: CALL GAMSDOC VMULFM (or @PRT IMSL*DOC.VMULFM) | Access: LIB NBS*IMSL
- VMULFP** Matrix multiplication of matrix a by the transpose of matrix b (full storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULFP (A,B,L,M,N,IA,IB,C,IC,IER) | On-line doc: CALL GAMSDOC VMULFP (or @PRT IMSL*DOC.VMULFP) | Access: LIB NBS*IMSL
- VMULFQ** Matrix multiplication (full by band symmetric matrices). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULFQ (A,M,N,IA,B,NC,IB,C,IC) | On-line doc: CALL GAMSDOC VMULFQ (or @PRT IMSL*DOC.VMULFQ) | Access: LIB NBS*IMSL
- VMULFS** Matrix multiplication (full by symmetric matrices). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULFS (A,B,L,M,IA,C,IC) | On-line doc: CALL GAMSDOC VMULFS (or @PRT IMSL*DOC.VMULFS) | Access: LIB NBS*IMSL
- VMULQB** Matrix multiplication (band symmetric by band matrices). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULQB (A,IA,B,IB,N,C,IC) | On-line doc: CALL GAMSDOC VMULQB (or @PRT IMSL*DOC.VMULQB) | Access: LIB NBS*IMSL
- VMULQF** Matrix multiplication (band symmetric by full matrices). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULQF (A,M,NC,IA,B,N,IB,C,IC) | On-line doc: CALL GAMSDOC VMULQF (or @PRT

IMSL*DOC.VMULQF) | Access: LIB NBS*IMSL

VMULQQ Matrix multiplication (band symmetric storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULQQ (A,N,NCA,IA,B,NCB,IB,C,IC) | On-line doc: CALL GAMSDOC VMULQQ (or @PRT IMSL*DOC.VMULQQ) | Access: LIB NBS*IMSL

VMULQS Matrix multiplication (band symmetric by symmetric matrices). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULQS (A,N,NC,IA,B,C,IC) | On-line doc: CALL GAMSDOC VMULQS (or @PRT IMSL*DOC.VMULQS) | Access: LIB NBS*IMSL

VMULSB Matrix multiplication (symmetric by band matrices). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULSB (A,B,IB,N,C,IC) | On-line doc: CALL GAMSDOC VMULSB (or @PRT IMSL*DOC.VMULSB) | Access: LIB NBS*IMSL

VMULSF Matrix multiplication (symmetric by full matrices). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULSF (A,N,B,M,IB,C,IC) | On-line doc: CALL GAMSDOC VMULSF (or @PRT IMSL*DOC.VMULSF) | Access: LIB NBS*IMSL

VMULSQ Matrix multiplication (symmetric by band symmetric matrices). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULSQ (A,N,B,NC,IB,C,IC) | On-line doc: CALL GAMSDOC VMULSQ (or @PRT IMSL*DOC.VMULSQ) | Access: LIB NBS*IMSL

VMULSS Matrix multiplication (symmetric storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULSS (A,B,N,C,IC) | On-line doc: CALL GAMSDOC VMULSS (or @PRT IMSL*DOC.VMULSS) | Access: LIB NBS*IMSL

VNRMF1 1-norm of matrices (full storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b2 | Usage: CALL VNRMF1 (A,N,IA,XNRMA) | On-line doc: CALL GAMSDOC VNRMF1 (or @PRT IMSL*DOC.VNRMF1) | Access: LIB NBS*IMSL

VNRMF2 Euclidean-norm of matrices (full storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b2 | Usage: CALL VNRMF2 (A,N,IA,XNRMA) | On-line doc: CALL GAMSDOC VNRMF2 (or @PRT IMSL*DOC.VNRMF2) | Access: LIB NBS*IMSL

VNRMF1 Infinity-norm matrices (full storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b2 | Usage: CALL VNRMF1 (A,N,IA,XNRMA) | On-line doc: CALL GAMSDOC VNRMF1 (or @PRT IMSL*DOC.VNRMF1) | Access: LIB NBS*IMSL

VNRMS1 1-norm of matrices (symmetric storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b2 | Usage: CALL VNRMS1 (A,N,XNRMA) | On-line doc: CALL GAMSDOC VNRMS1 (or @PRT IMSL*DOC.VNRMS1) | Access: LIB NBS*IMSL

VNRMS2 Euclidean-norm of matrices (symmetric storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b2 | Usage: CALL VNRMS2 (A,N,XNRMA) | On-line doc: CALL GAMSDOC VNRMS2 (or @PRT IMSL*DOC.VNRMS2) | Access: LIB NBS*IMSL

VPLT Displays an Nx100 character line printer plot of the N values of a series (horizontal axis) vs. their indices (vertical axis). | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c5 Q1 | Usage: CALL VPLT (Y, N, NPLOT) | On-line doc: CALL GAMSDOC VPLT (or @PRT STATLIB*DOC.VPLT) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

VPLT2 Displays an Nx100 character line printer plot of the N values of each of two series (horizontal axis) vs. their indices (vertical axis). | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c5 Q1 | Usage: CALL VPLT2 (Y1, Y2, N, NPLOT, NSCALE) | On-line doc: CALL GAMSDOC VPLT2 (or @PRT STATLIB*DOC.VPLT2) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

VPLT2L Displays an Nx100 character line printer plot of the N values of each of two series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c5 Q1 | Usage: CALL VPLT2L (Y1, Y2, N, NPLOT, NSCALE, LMSPEC, Y1LB, Y1UB, Y2LB, Y2UB) | On-line doc: CALL GAMSDOC VPLT2L (or @PRT STATLIB*DOC.VPLT2L) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

VPLTB Displays an Nx100 character line printer bar plot of the N values of a series (horizontal axis) vs. their indices (vertical axis). | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c5 Q1 | Usage: CALL VPLTB (Y, N, NPLOT) | On-line doc: CALL GAMSDOC VPLTB (or @PRT STATLIB*DOC.VPLTB) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

VPLTBL Displays an Nx100 character line printer bar plot of the N values of a series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c5 Q1 | Usage: CALL VPLTBL (Y, N, NPLOT, YLB, YUB) | On-line doc: CALL GAMSDOC VPLTBL (or @PRT STATLIB*DOC.VPLTBL) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

VPLTL Displays an Nx100 character line printer plot of the N values of a series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c5 Q1 | Usage: CALL VPLTL (Y, N, NPLOT, YLB, YUB) | On-line doc: CALL GAMSDOC VPLTL (or @PRT STATLIB*DOC.VPLTL) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

- VPOLYF** Matrix polynomial (full storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b7 | Usage: CALL VPOLYF (A,N,IA,COEF,M,B,IB,WKAREA) | On-line doc: CALL GAMSDOC VPOLYF (or @PRT IMSL*DOC.VPOLYF) | Access: LIB NBS*IMSL
- VSAR** Sorting of matrices (with options). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N6a2b1 N6a1b1 | Usage: CALL VSAR(A,IA,NR,NC,IOP,KPOS,NK,IR,WK,IER) | On-line doc: CALL GAMSDOC VSAR (or @PRT IMSL*DOC.VSAR) | Access: LIB NBS*IMSL
- VSODA** Sorting of columns of a double precision matrix in ascending order of keys in rows. | Proprietary double precision Fortran subprogram in IMSL library. | Class(es): N6a2b2 | Usage: CALL VSODA(A,IA,NR,NC,NK,WK,IER) | On-line doc: CALL GAMSDOC VSODA (or @PRT IMSL*DOC.VSODA) | Access: LIB NBS*IMSL
- VSORA** Sorting of columns of a real matrix into ascending order of keys in rows. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N6a2b1 | Usage: CALL VSORA(A,IA,NR,NC,NK,WK,IER) | On-line doc: CALL GAMSDOC VSORA (or @PRT IMSL*DOC.VSORA) | Access: LIB NBS*IMSL
- VSRTA** Sorting of arrays by algebraic value. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N6a2b1 | Usage: CALL VSRTA (A,LA) | On-line doc: CALL GAMSDOC VSRTA (or @PRT IMSL*DOC.VSRTA) | Access: LIB NBS*IMSL
- VSRTM** Sorting of arrays by absolute value. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N6a2b1 | Usage: CALL VSRTM (A,LA) | On-line doc: CALL GAMSDOC VSRTM (or @PRT IMSL*DOC.VSRTM) | Access: LIB NBS*IMSL
- VS RTP** Sorting of arrays by absolute value permutations returned. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N6a2b1 | Usage: CALL VS RTP (A,LA,IR) | On-line doc: CALL GAMSDOC VS RTP (or @PRT IMSL*DOC.VS RTP) | Access: LIB NBS*IMSL
- VS RTR** Sorting of arrays by algebraic value permutations returned. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N6a2b1 | Usage: CALL VS RTR (A,LA,IR) | On-line doc: CALL GAMSDOC VS RTR (or @PRT IMSL*DOC.VS RTR) | Access: LIB NBS*IMSL
- VS RTU** Interchange the rows or columns of a matrix using a permutation vector such as the one obtained from IMSL routines VS RTP or VS RTR. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N8 | Usage: CALL VS RTU (Z,IZ,N,M,IND,IR,WK) | On-line doc: CALL GAMSDOC VS RTU (or @PRT IMSL*DOC.VS RTU) | Access: LIB NBS*IMSL
- VT PROF** Transpose product of matrix (full storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VT PROF (A,L,M,IA,ATA) | On-line doc: CALL GAMSDOC VT PROF (or @PRT IMSL*DOC.VT PROF) | Access: LIB NBS*IMSL
- VT PROS** Transpose product of a matrix (symmetric storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VT PROS (A,N,ATA) | On-line doc: CALL GAMSDOC VT PROS (or @PRT IMSL*DOC.VT PROS) | Access: LIB NBS*IMSL
- VTRAN** Transpose a rectangular matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b3 | Usage: CALL VTRAN (A,N,M) | On-line doc: CALL GAMSDOC VTRAN (or @PRT IMSL*DOC.VTRAN) | Access: LIB NBS*IMSL
- VUABQ** Matrix addition (band + band symmetric matrices). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b5 | Usage: CALL VUABQ (A,IA,B,IB,N,C,IC) | On-line doc: CALL GAMSDOC VUABQ (or @PRT IMSL*DOC.VUABQ) | Access: LIB NBS*IMSL
- VUAFB** Matrix addition (full + band matrices). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b5 | Usage: CALL VUAFB (A,IA,B,IB,N,C,IC) | On-line doc: CALL GAMSDOC VUAFB (or @PRT IMSL*DOC.VUAFB) | Access: LIB NBS*IMSL
- VUAFQ** Matrix addition (full + band symmetric matrices). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b5 | Usage: CALL VUAFQ (A,N,IA,B,NC,IB,C,IC) | On-line doc: CALL GAMSDOC VUAFQ (or @PRT IMSL*DOC.VUAFQ) | Access: LIB NBS*IMSL
- VUAFS** Matrix addition (full + symmetric matrices). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b5 | Usage: CALL VUAFS (A,N,IA,B,C,IC) | On-line doc: CALL GAMSDOC VUAFS (or @PRT IMSL*DOC.VUAFS) | Access: LIB NBS*IMSL
- VUASB** Matrix addition (symmetric + band matrices). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b5 | Usage: CALL VUASB (A,B,IB,N,C,IC) | On-line doc: CALL GAMSDOC VUASB (or @PRT IMSL*DOC.VUASB) | Access: LIB NBS*IMSL
- VUASQ** Matrix addition (symmetric + band symmetric matrices). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b5 | Usage: CALL VUASQ (A,N,B,NC,IB,C) | On-line doc: CALL GAMSDOC VUASQ (or @PRT IMSL*DOC.VUASQ) | Access: LIB NBS*IMSL

W

- WALSH** Calculates $(X(i) + X(j))/2$ and stores these average and their indices (useful for nonparametric tests and confidence intervals). | Command in MINITAB Proprietary interactive system. Class(es): L2a | Usage: WALSh averages of values in C, put into C [indices into C and C] | On-line doc: HELP WALSH (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- WEIB** Performs a Weibull distribution analysis on the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L4a1a23 | Usage: CALL WEIB(X,N) | On-line doc: CALL GAMSDOC WEIB (or @PRT DATAPAC*DOC.WEIB) | Access: LIB NBS*DATAPAC
- WEICDF** Computes the cumulative distribution function value for the Weibull distribution with tail length parameter GAMMA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1w | Usage: CALL WEICDF(X,GAMMA,CDF) | On-line doc: CALL GAMSDOC WEICDF (or @PRT DATAPAC*DOC.WEICDF) | Access: LIB NBS*DATAPAC
- WEIPLT** Generates a Weibull probability plot with tail length parameter GAMMA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c4w | Usage: CALL WEIPLT(X,N,GAMMA) | On-line doc: CALL GAMSDOC WEIPLT (or @PRT DATAPAC*DOC.WEIPLT) | Access: LIB NBS*DATAPAC
- WEIPPF** Computes the percent point function value for the Weibull distribution with tail length parameter GAMMA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2w | Usage: CALL WEIPPF(P,GAMMA,PPF) | On-line doc: CALL GAMSDOC WEIPPF (or @PRT DATAPAC*DOC.WEIPPF) | Access: LIB NBS*DATAPAC
- WEIRAN** Generates a random sample of size N from the Weibull distribution with tail length parameter GAMMA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a23 | Usage: CALL WEIRAN(N,GAMMA,ISTART,X) | On-line doc: CALL GAMSDOC WEIRAN (or @PRT DATAPAC*DOC.WEIRAN) | Access: LIB NBS*DATAPAC
- WIND** Computes the sample Windsorized mean of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1a | Usage: CALL WIND(X,N,P,P1,P2,IWRITE,XWIND) | On-line doc: CALL GAMSDOC WIND (or @PRT DATAPAC*DOC.WIND) | Access: LIB NBS*DATAPAC
- WINTERVAL** Calculates a one-sample Wilcoxon rank estimate and confidence interval for the center of a symmetric distribution. | Command in MINITAB Proprietary interactive system. Class(es): L4a1b | Usage: WINTerval [percent confidence K] for data in C,...,C | On-line doc: HELP WINTERVAL (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- WNNLS** Solves linearly constrained non-negative least squares problem. | Portable single precision Fortran subprogram in FC sublibrary of CMLIB library. | Class(es): K1a2a | Usage: CALL WNNLS(W,MDW,ME,MA,N,L,PRGOPT,X,RNORM,MODE,IWORK,WORK) | On-line doc: CALL GAMSDOC WNNLS (or @PRT CMLIB*DOC.WNNLS/FC) | Tests: CMLIB*TEST-SOURCE.\$F/FC | Access: LIB NBS*CMLIB
- WRITE** Writes out the contents of the vector X in an orderly and neat fashion. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): N1 | Usage: CALL WRITE(X,N,NNLINE,IWIDTH,IDEC) | On-line doc: CALL GAMSDOC WRITE (or @PRT DATAPAC*DOC.WRITE) | Access: LIB NBS*DATAPAC
- WTEST** Performs one-sample one- or two-sided Wilcoxon signed-rank tests. | Command in MINITAB Proprietary interactive system. Class(es): L4a1b | Usage: WTESt [of center = K] on data in C,...,C [; subcommand ALTERNATIVE = K.] | On-line doc: HELP WTEST (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

X

- X01AAE** Pi. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X01AAF. | Class(es): R1 | Usage: D = X01AAE (X) | On-line doc: CALL GAMSDOC X01AAE (or @PRT NAG*DOC.X01AAE) | Access: LIB NBS*NAG
- X01AAF** Pi. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X01AAE. | Class(es): R1 | Usage: D = X01AAF (X) | On-line doc: CALL GAMSDOC X01AAF (or @PRT NAG*DOC.X01AAF) | Access: LIB NBS*NAG
- X01ABE** Euler's constant, gamma. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X01ABF. | Class(es): R1 | Usage: D = X01ABE (X) | On-line doc: CALL GAMSDOC X01ABE (or @PRT NAG*DOC.X01ABE) | Access: LIB NBS*NAG
- X01ABF** Euler's constant, gamma. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X01ABE. | Class(es): R1 | Usage: D = X01ABF (X) | On-line doc: CALL GAMSDOC X01ABF (or @PRT NAG*DOC.X01ABF) | Access: LIB NBS*NAG
- X02AAE** Smallest possible e such that $1.0+e > 1.0$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02AAF. | Class(es): R1 | Usage: D = X02AAE (X) | On-line doc: CALL GAMSDOC X02AAE (or @PRT NAG*DOC.X02AAE) | Access: LIB NBS*NAG
- X02AAF** Smallest possible e such that $1.0+e > 1.0$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02AAE. | Class(es): R1 | Usage: D = X02AAF (X) | On-line doc: CALL GAMSDOC X02AAF (or @PRT NAG*DOC.X02AAF) | Access: LIB NBS*NAG
- X02ABE** Smallest representable positive real number. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02ABF. | Class(es): R1 | Usage: D = X02ABE (X) | On-line doc: CALL GAMSDOC X02ABE (or @PRT NAG*DOC.X02ABE) | Access: LIB NBS*NAG
- X02ABF** Smallest representable positive real number. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02ABE. | Class(es): R1 | Usage: D = X02ABF (X) | On-line doc: CALL GAMSDOC X02ABF (or @PRT NAG*DOC.X02ABF) | Access: LIB NBS*NAG
- X02ACE** Largest representable positive real number. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02ACF. | Class(es): R1 | Usage: D = X02ACE (X) | On-line doc: CALL GAMSDOC X02ACE (or @PRT NAG*DOC.X02ACE) | Access: LIB NBS*NAG
- X02ACF** Largest representable positive real number. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02ACE. | Class(es): R1 | Usage: D = X02ACF (X) | On-line doc: CALL GAMSDOC X02ACF (or @PRT NAG*DOC.X02ACF) | Access: LIB NBS*NAG
- X02ADE** Ratio of X02ABE to X02AAE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02ADF. | Class(es): R1 | Usage: D = X02ADE (X) | On-line doc: CALL GAMSDOC X02ADE (or @PRT NAG*DOC.X02ADE) | Access: LIB NBS*NAG
- X02ADF** Ratio of X02ABF to X02AAF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02ADE. | Class(es): R1 | Usage: D = X02ADF (X) | On-line doc: CALL GAMSDOC X02ADF (or @PRT NAG*DOC.X02ADF) | Access: LIB NBS*NAG
- X02AEE** Largest negative permissible argument for exp. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02AEF. | Class(es): R1 | Usage: D = X02AEE (X) | On-line doc: CALL GAMSDOC X02AEE (or @PRT NAG*DOC.X02AEE) | Access: LIB NBS*NAG
- X02AEF** Largest negative permissible argument for exp. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02AEE. | Class(es): R1 | Usage: D = X02AEF (X) | On-line doc: CALL GAMSDOC X02AEF (or @PRT NAG*DOC.X02AEF) | Access: LIB NBS*NAG
- X02AFE** Returns the value of the largest positive argument permitted for EXP. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02AFF. | Class(es): R1 | Usage: R = X02AFE(X) | On-line doc: CALL GAMSDOC X02AFE (or @PRT NAG*DOC.X02AFE) | Access: LIB NBS*NAG
- X02AFF** Returns the value of the largest positive argument permitted for DEXP. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02AFE. | Class(es): R1 | Usage: D = X02AFF(X) | On-line doc: CALL GAMSDOC X02AFF (or @PRT NAG*DOC.X02AFF) | Access: LIB NBS*NAG
- X02AGE** Smallest representable positive real number with representable inverse. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02AGF. | Class(es): R1 | Usage: I = X02AGE (X) | On-line doc: CALL GAMSDOC X02AGE (or @PRT NAG*DOC.X02AGE) | Access: LIB NBS*NAG
- X02AGF** Smallest representable positive real number with representable inverse. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02AGE. | Class(es): R1 | Usage: I = X02AGF (X) | On-line doc: CALL GAMSDOC X02AGF (or @PRT NAG*DOC.X02AGF) | Access: LIB NBS*NAG

- X02AHE** Returns the value of the largest positive argument permitted for SIN and COS. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02AHF. | Class(es): R1 | Usage: R = X02AHE(X) | On-line doc: CALL GAMSDOC X02AHE (or @PRT NAG*DOC.X02AHE) | Access: LIB NBS*NAG
- X02AHF** Returns the value of the largest positive argument permitted for DSIN and DCOS. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02AHE. | Class(es): R1 | Usage: D = X02AHF(X) | On-line doc: CALL GAMSDOC X02AHF (or @PRT NAG*DOC.X02AHF) | Access: LIB NBS*NAG
- X02BAE** Base of floating-point arithmetic. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02BAF. | Class(es): R1 | Usage: 1 = X02BAE (X) | On-line doc: CALL GAMSDOC X02BAE (or @PRT NAG*DOC.X02BAE) | Access: LIB NBS*NAG
- X02BAF** Base of floating-point arithmetic. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02BAE. | Class(es): R1 | Usage: 1 = X02BAF (X) | On-line doc: CALL GAMSDOC X02BAF (or @PRT NAG*DOC.X02BAF) | Access: LIB NBS*NAG
- X02BBE** Largest representable integer. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02BBF. | Class(es): R1 | Usage: 1 = X02BBE (X) | On-line doc: CALL GAMSDOC X02BBE (or @PRT NAG*DOC.X02BBE) | Access: LIB NBS*NAG
- X02BBF** Largest representable integer. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02BBE. | Class(es): R1 | Usage: 1 = X02BBF (X) | On-line doc: CALL GAMSDOC X02BBF (or @PRT NAG*DOC.X02BBF) | Access: LIB NBS*NAG
- X02BCE** Largest positive integer power to which 2.0 can be raised without overflow. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02BCF. | Class(es): R1 | Usage: 1 = X02BCE (X) | On-line doc: CALL GAMSDOC X02BCE (or @PRT NAG*DOC.X02BCE) | Access: LIB NBS*NAG
- X02BCF** Largest positive integer power to which 2.0 can be raised without overflow. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02BCE. | Class(es): R1 | Usage: 1 = X02BCF (X) | On-line doc: CALL GAMSDOC X02BCF (or @PRT NAG*DOC.X02BCF) | Access: LIB NBS*NAG
- X02BDE** Largest negative integer power to which 2.0 can be raised without underflow. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02BDF. | Class(es): R1 | Usage: 1 = X02BDE (X) | On-line doc: CALL GAMSDOC X02BDE (or @PRT NAG*DOC.X02BDE) | Access: LIB NBS*NAG
- X02BDF** Largest negative integer power to which 2.0 can be raised without underflow. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02BDE. | Class(es): R1 | Usage: 1 = X02BDF (X) | On-line doc: CALL GAMSDOC X02BDF (or @PRT NAG*DOC.X02BDF) | Access: LIB NBS*NAG
- X02BEE** Maximum number of decimal digits that can be represented. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02BEF. | Class(es): R1 | Usage: 1 = X02BEE (X) | On-line doc: CALL GAMSDOC X02BEE (or @PRT NAG*DOC.X02BEE) | Access: LIB NBS*NAG
- X02BEF** Maximum number of decimal digits that can be represented. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02BEE. | Class(es): R1 | Usage: 1 = X02BEF (X) | On-line doc: CALL GAMSDOC X02BEF (or @PRT NAG*DOC.X02BEF) | Access: LIB NBS*NAG
- X02CAE** Estimate of active-set size (on machines with paged virtual store). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02CAF. | Class(es): R1 | Usage: 1 = X02CAE (X) | On-line doc: CALL GAMSDOC X02CAE (or @PRT NAG*DOC.X02CAE) | Access: LIB NBS*NAG
- X02CAF** Estimate of active-set size (on machines with paged virtual store). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02CAE. | Class(es): R1 | Usage: 1 = X02CAF (X) | On-line doc: CALL GAMSDOC X02CAF (or @PRT NAG*DOC.X02CAF) | Access: LIB NBS*NAG
- X02DAE** Switch for taking precautions to avoid underflow. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02DAF. | Class(es): R1 | Usage: L = X02DAE (X) | On-line doc: CALL GAMSDOC X02DAE (or @PRT NAG*DOC.X02DAE) | Access: LIB NBS*NAG
- X02DAF** Switch for taking precautions to avoid underflow. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02DAE. | Class(es): R1 | Usage: L = X02DAF (X) | On-line doc: CALL GAMSDOC X02DAF (or @PRT NAG*DOC.X02DAF) | Access: LIB NBS*NAG
- X03AAE** Real innerproduct added to initial value, basic/additional precision. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X03AAF. | Class(es): D1a4 | Usage: CALL X03AAE (A, ISIZEA, B, ISIZEB, N, ISTEPA, ISTEPB, C1, C2, D1, D2, SW, IFAIL) | On-line doc: CALL GAMSDOC X03AAE (or @PRT NAG*DOC.X03AAE) | Access: LIB NBS*NAG
- X03AAF** Real innerproduct added to initial value, basic/additional precision. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X03AAE. | Class(es): D1a4 | Usage: CALL X03AAF (A, ISIZEA, B, ISIZEB, N, ISTEPA, ISTEPB, C1, C2, D1, D2, SW, IFAIL) | On-line doc: CALL GAMSDOC X03AAF (or @PRT NAG*DOC.X03AAF) | Access: LIB NBS*NAG

- C1, C2, D1, D2, SW, IFAIL) | On-line doc: CALL GAMSDOC X03AAF (or @PRT NAG*DOC.X03AAF) | Access: LIB NBS*NAG
- X03ABE** Complex innerproduct added to initial value, basic/additional precision. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X03ABF. | Class(es): D1a4 | Usage: CALL X03ABE (A, ISIZEA, B, ISIZEB, N, ISTEPA, ISTEPB, CX, DX, SW, IFAIL) | On-line doc: CALL GAMSDOC X03ABE (or @PRT NAG*DOC.X03ABE) | Access: LIB NBS*NAG
- X03ABF** Complex innerproduct added to initial value, basic/additional precision. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X03ABE. | Class(es): D1a4 | Usage: CALL X03ABF (A, ISIZEA, B, ISIZEB, N, ISTEPA, ISTEPB, CX, DX, SW, IFAIL) | On-line doc: CALL GAMSDOC X03ABF (or @PRT NAG*DOC.X03ABF) | Access: LIB NBS*NAG
- X04AAE** Return or set unit number for error messages for Nag library programs. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X04AAF. | Class(es): R3b | Usage: CALL X04AAE (IFLAG, NERR) | On-line doc: CALL GAMSDOC X04AAE (or @PRT NAG*DOC.X04AAE) | Access: LIB NBS*NAG
- X04AAF** Return or set unit number for error messages for Nag library programs. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X04AAE. | Class(es): R3b | Usage: CALL X04AAF (IFLAG, NERR) | On-line doc: CALL GAMSDOC X04AAF (or @PRT NAG*DOC.X04AAF) | Access: LIB NBS*NAG
- X04ABE** Return or set unit number for advisory messages for NAG library programs. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X04ABF. | Class(es): R3b | Usage: CALL X04ABE (IFLAG, NADV) | On-line doc: CALL GAMSDOC X04ABE (or @PRT NAG*DOC.X04ABE) | Access: LIB NBS*NAG
- X04ABF** Return or set unit number for advisory messages for NAG library programs. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X04ABE. | Class(es): R3b | Usage: CALL X04ABF (IFLAG, NADV) | On-line doc: CALL GAMSDOC X04ABF (or @PRT NAG*DOC.X04ABF) | Access: LIB NBS*NAG
- XDLEGF** Calculates sequences of values of Legendre functions of the first kind of positive or negative order, as well as Legendre functions of the second kind, and normalized Legendre functions. | Portable double precision Fortran subprogram in FCNPAK sublibrary of CMLIB library. Single precision version is XSLEGF. | Class(es): C0 C3a2 | Usage: CALL XDLEGF(DNU1, NUDIFF, MU1, MU2, THETA, ID, PQA, IPQA) | On-line doc: CALL GAMSDOC XDLEGF (or @PRT CMLIB*DOC.XDLEGF/FCNPAK) | Tests: CMLIB*TEST-SOURCE.XDLEGF/FCNPAK | Access: LIB NBS*CMLIB | See also: XDSET
- XDNRMP** Calculates a sequence of values of the normalized Legendre polynomials for fixed degree and argument and variable order. | Portable double precision Fortran subprogram in FCNPAK sublibrary of CMLIB library. Single precision version is XSNRMP. | Class(es): C0 C3a2 | Usage: CALL XDNRMP(NU, MU1, MU2, ARG, MODE, PN, IPN, ISIG) | On-line doc: CALL GAMSDOC XDNRMP (or @PRT CMLIB*DOC.XDNRMP/FCNPAK) | Tests: CMLIB*TEST-SOURCE.XDNRMP/FCNPAK | Access: LIB NBS*CMLIB | See also: XDSET
- XERABT** Terminate run and print traceback. (Requires system dependent programming to execute properly, else just STOPs.) | Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XERABT(MESSG, NMESSG) | On-line doc: CALL GAMSDOC XERABT (or @PRT CMLIB*DOC.SUMMARY/XERROR) | Access: LIB NBS*CMLIB
- XERCLR** Clear current message number. | Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XERCLR | On-line doc: CALL GAMSDOC XERCLR (or @PRT CMLIB*DOC.SUMMARY/XERROR) | Access: LIB NBS*CMLIB
- XERCTL** Perform special error processing of one message. | Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XERCTL(MESSG1, NMESSG, NERR, LEVEL, KONTRL) | On-line doc: CALL GAMSDOC XERCTL (or @PRT CMLIB*DOC.SUMMARY/XERROR) | Access: LIB NBS*CMLIB
- XERDMP** Print error summary and clear tables. | Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XERDMP | On-line doc: CALL GAMSDOC XERDMP (or @PRT CMLIB*DOC.SUMMARY/XERROR) | Access: LIB NBS*CMLIB
- XERMAX** Set limit of MAX times each message can be printed. | Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XERMAX(MAX) | On-line doc: CALL GAMSDOC XERMAX (or @PRT CMLIB*DOC.SUMMARY/XERROR) | Access: LIB NBS*CMLIB
- XERROR** Process a message. | Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XERROR(MESSG, NMESSG, NERR, LEVEL) | On-line doc: CALL GAMSDOC XERROR (or @PRT CMLIB*DOC.SUMMARY/XERROR) | Access: LIB NBS*CMLIB
- XERRWV** Process a message with numeric values. | Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XERRWV(MESSG, NMESSG, NERR, LEVEL, N1, I1, I2, NR, R1, R2) | On-line doc: CALL GAMSDOC XERRWV (or @PRT CMLIB*DOC.SUMMARY/XERROR) | Access: LIB NBS*CMLIB
- XGETF** Get current value of KONTRL. | Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XGETF(KONTRL) | On-line doc: CALL GAMSDOC XGETF (or @PRT CMLIB*DOC.SUMMARY/XERROR) | Access: LIB NBS*CMLIB

- XGETUA** Get current output unit numbers. | Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XGETUA(IUNITA,N) | On-line doc: CALL GAMSDOC XGETUA (or @PRT CMLIB*DOC.SUMMARY/XERROR) | Access: LIB NBS*CMLIB
- XGETUN** Get current output unit number. | Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XGETUN(IUNIT) | On-line doc: CALL GAMSDOC XGETUN (or @PRT CMLIB*DOC.SUMMARY/XERROR) | Access: LIB NBS*CMLIB
- XSETF** Set KONTRL for XERROR, default is =2. | Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3a | Usage: CALL XSETF(KONTRL) | On-line doc: CALL GAMSDOC XSETF (or @PRT CMLIB*DOC.SUMMARY/XERROR) | Access: LIB NBS*CMLIB
- XSETUA** Set up to 5 output unit numbers. | Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3b | Usage: CALL XSETUA(IUNITA,N) | On-line doc: CALL GAMSDOC XSETUA (or @PRT CMLIB*DOC.SUMMARY/XERROR) | Access: LIB NBS*CMLIB
- XSETUN** Set one output unit number. | Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3b | Usage: CALL XSETUN(IUNIT) | On-line doc: CALL GAMSDOC XSETUN (or @PRT CMLIB*DOC.SUMMARY/XERROR) | Access: LIB NBS*CMLIB
- XSLEGF** Calculates sequences of values of Legendre functions of the first kind of positive or negative order, as well as Legendre functions of the second kind, and normalized Legendre functions. | Portable single precision Fortran subprogram in FCNPAK sublibrary of CMLIB library. Double precision version is XDLEGF. | Class(es): C9 C3a2 | Usage: CALL XSLEGF(DNU1,NUDIFF,MU1,MU2,THETA,ID,PQA,IPQA) | On-line doc: CALL GAMSDOC XSLEGF (or @PRT CMLIB*DOC.XSLEGF/FCNPAK) | Tests: CMLIB*TEST-SOURCE.XSLEGF/FCNPAK | Access: LIB NBS*CMLIB | See also: XSSET
- XSNRMP** Calculates a sequence of values of the normalized Legendre polynomials for fixed degree and argument and variable order. | Portable single precision Fortran subprogram in FCNPAK sublibrary of CMLIB library. Double precision version is XDNRMP. | Class(es): C9 C3a2 | Usage: CALL XSNRMP(NU,MU1,MU2,ARG,MODE,PN,IPN,ISIG) | On-line doc: CALL GAMSDOC XSNRMP (or @PRT CMLIB*DOC.XSNRMP/FCNPAK) | Tests: CMLIB*TEST-SOURCE.XSNRMP/FCNPAK | Access: LIB NBS*CMLIB | See also: XSSET

Z

- ZANLYT** Zeros of an analytic complex function using the Muller method with deflation. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): F1a2 | Usage: CALL ZANLYT (F,EPS,NSIG,KN,NGUESS,N,X,ITMAX,INFER,IER) | On-line doc: CALL GAMSDOC ZANLYT (or @PRT IMSL*DOC.ZANLYT) | Access: LIB NBS*IMSL
- ZBRENT** Zero of a function which changes sign in a given interval (Brent algorithm). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): F1b | Usage: CALL ZBRENT (F,EPS,NSIG,A,B,MAXFN,IER) | On-line doc: CALL GAMSDOC ZBRENT (or @PRT IMSL*DOC.ZBRENT) | Access: LIB NBS*IMSL
- ZCPOLY** Zeros of a polynomial with complex coefficients (Jenkins-Traub). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): F1a1b | Usage: CALL ZCPOLY (A,NDEG,Z,IER) | On-line doc: CALL GAMSDOC ZCPOLY (or @PRT IMSL*DOC.ZCPOLY) | Access: LIB NBS*IMSL
- ZERO** Finds a single real root of a function within an interval specified by the user. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DZERO. | Class(es): F1b | Usage: X = ZERO (F,A,B,T) | On-line doc: CALL GAMSDOC ZERO (or @PRT PORT*DOC.ZERO) | Access: LIB NBS*PORT
- ZEROIN** Finds a zero of a user defined function on an interval given the endpoints A and B such that $F(A)*F(B) < 0$. | Portable single precision Fortran subprogram in ZEROIN sublibrary of CMLIB library. | Class(es): F1b | Usage: CALL ZEROIN (F,B,C,RE,AE,IFLAG) | On-line doc: CALL GAMSDOC ZEROIN (or @PRT CMLIB*DOC.ZEROIN/ZEROIN) | Tests: CMLIB*TEST-SOURCE.ZEROIN/ZEROIN | Access: LIB NBS*CMLIB
- ZFALSE** Zero of a function given an interval containing the zero. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): F1a2 | Usage: CALL ZFALSE (F,EPS,NSIG,XL,XR,XAPP,ITMAX,IER) | On-line doc: CALL GAMSDOC ZFALSE (or @PRT IMSL*DOC.ZFALSE) | Access: LIB NBS*IMSL
- ZINTERVAL** Calculates a z-confidence interval with specified percent confidence and standard deviation. | Command in MINITAB Proprietary interactive system. Class(es): L4a1a14 L4a2 | Usage: ZINTERVAL K percent confidence, assuming $\sigma = K$, on C | On-line doc: HELP ZINTERVAL (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
- ZONE** Finds a solution of a system of non-linear equations. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DZONE. | Class(es): F2a | Usage: CALL ZONE (FUNC, N, X, EPS, JMAX, F2NORM) | On-line doc: CALL GAMSDOC ZONE (or @PRT PORT*DOC.ZONE) | Access: LIB NBS*PORT
- ZONEJ** Finds a solution of a system of non-linear equations. User must provide a SUBROUTINE to compute the Jacobian matrix. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DZONEJ. | Class(es): F2a | Usage: CALL ZONEJ (FUNC, Z1JAC, N, X, EPS, JMAX, F2NORM) | On-line doc: CALL GAMSDOC ZONEJ (or @PRT PORT*DOC.ZONEJ) | Access: LIB NBS*PORT
- ZPOLR** Zeros of a polynomial with real coefficients (Laguerre). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): F1a1a | Usage: CALL ZPOLR (A,NDEG,Z,IER) | On-line doc: CALL GAMSDOC ZPOLR (or @PRT IMSL*DOC.ZPOLR) | Access: LIB NBS*IMSL
- ZQADC** Zeros of a quadratic with complex coefficients. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): F1a1b | Usage: CALL ZQADC (A,B,C,ZSM,ZLG,IER) | On-line doc: CALL GAMSDOC ZQADC (or @PRT IMSL*DOC.ZQADC) | Access: LIB NBS*IMSL
- ZQADR** Zeros of a quadratic with real coefficients. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): F1a1a | Usage: CALL ZQADR (A,B,C,ZSM,ZLG,IER) | On-line doc: CALL GAMSDOC ZQADR (or @PRT IMSL*DOC.ZQADR) | Access: LIB NBS*IMSL
- ZREAL1** The real zeros of a real function - to be used when initial guesses are poor. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): F1a2 | Usage: CALL ZREAL1 (F,EPS,EPS2,ETA,NSIG,N,X,ITMAX,IER) | On-line doc: CALL GAMSDOC ZREAL1 (or @PRT IMSL*DOC.ZREAL1) | Access: LIB NBS*IMSL
- ZREAL2** The real zeros of a real function - to be used when initial guesses are good. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): F1a2 | Usage: CALL ZREAL2 (F,EPS,EPS2,ETA,NSIG,N,X,ITMAX,IER) | On-line doc: CALL GAMSDOC ZREAL2 (or @PRT IMSL*DOC.ZREAL2) | Access: LIB NBS*IMSL
- ZRPOLY** Zeros of a polynomial with real coefficients (Jenkins-Traub). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): F1a1a | Usage: CALL ZRPOLY (A,NDEG,Z,IER) | On-line doc: CALL GAMSDOC ZRPOLY (or @PRT IMSL*DOC.ZRPOLY) | Access: LIB NBS*IMSL
- ZSCNT** Solve a system of nonlinear equations. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): F2a | Usage: CALL ZSCNT (FCN,NSIG,N,ITMAX,PAR,X,FNORM,WK,IER) | On-line doc: CALL GAMSDOC ZSCNT (or @PRT IMSL*DOC.ZSCNT) | Access: LIB NBS*IMSL
- ZSPOW** Solve a system of nonlinear equations (uses function values only). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): F2a | Usage: CALL ZSPOW (FCN,NSIG,N,ITMAX,PAR,X,FNORM,WK,IER) | On-line doc: CALL GAMSDOC ZSPOW (or

@PRT IMSL*DOC.ZSPOW) | Access: LIB NBS*IMSL

ZSRCH Generate points in an n dimensional space. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G4d F3 L6b
| Usage: CALL ZSRCH (A,B,N,K,IP,S,M,IW,IER) | On-line doc: CALL GAMSDOC ZSRCH (or @PRT IMSL*DOC.ZSRCH) | Access: LIB NBS*IMSL

ZTEST Performs a one- or two-sided z-test for a specified standard deviation. | Command in MINITAB Proprietary interactive system.
Class(es): L4a1a14 L4a2 | Usage: ZTEST of $\mu = K$ assuming $\sigma = K$, on C [; subcommand ALTERNATIVE = K.] | On-line doc: HELP ZTEST (in Minitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

ZX0LP Solve the linear programming problem (phase one or phase two) via the revised simplex algorithm. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G2a1 | Usage: CALL ZX0LP (IPHASE,C,D,ICOLMS,ROW,K,M,N,ITMAX,LIC,IR,COPI,IDES,X,WA,IER) | On-line doc: CALL GAMSDOC ZX0LP (or @PRT IMSL*DOC.ZX0LP) | Access: LIB NBS*IMSL

ZX3LP Solve the linear programming problem via the revised simplex algorithm easy to use version. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G2a1 | Usage: CALL ZX3LP (A,IA,B,C,N,M1,M2,S,PSOL,DSOL,RW,IW,IER) | On-line doc: CALL GAMSDOC ZX3LP (or @PRT IMSL*DOC.ZX3LP) | Access: LIB NBS*IMSL

ZX4LP Solve the linear programming problem via the revised simplex algorithm (alternate easy to use version). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G2a1 | Usage: CALL ZX4LP (A,IA,B,C,N,M1,M2,S,PSOL,DSOL,RW,IW,IER) | On-line doc: CALL GAMSDOC ZX4LP (or @PRT IMSL*DOC.ZX4LP) | Access: LIB NBS*IMSL

ZXCGR A conjugate gradient algorithm for finding the minimum of a function of n variables. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G1b1a | Usage: CALL ZXCGR (FUNCT,N,ACC,MAXFN,DFPRED,X,G,F,W,IER) | On-line doc: CALL GAMSDOC ZXCGR (or @PRT IMSL*DOC.ZXCGR) | Access: LIB NBS*IMSL

ZXGSN One-dimensional unimodal function minimization using the golden section search method. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G1a2 | Usage: CALL ZXGSN (F,A,B,TOL,XMIN,IER) | On-line doc: CALL GAMSDOC ZXGSN (or @PRT IMSL*DOC.ZXGSN) | Access: LIB NBS*IMSL

ZXGSP One-dimensional unimodal function minimization using the golden section search method - data parameters specified. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G1a2 | Usage: CALL ZXGSP (F,P1,P2,IP3,IP4,IP5,A,B,TOL,XMIN,IER) | On-line doc: CALL GAMSDOC ZXGSP (or @PRT IMSL*DOC.ZXGSP) | Access: LIB NBS*IMSL

ZXLSF One-dimensional minimization of a smooth function using safeguarded quadratic interpolation. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G1a1a | Usage: CALL ZXLSF(FUNC,X,STEP,BOUND,XACC,MAXFN,IER) | On-line doc: CALL GAMSDOC ZXLSF (or @PRT IMSL*DOC.ZXLSF) | Access: LIB NBS*IMSL

ZXMIN Minimum of a function of n variables using a quasi-Newton method. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G1b1a | Usage: CALL ZXMIN (FUNCT,N,NSIG,MAXFN,IOPT,X,H,G,F,W,IER) | On-line doc: CALL GAMSDOC ZXMIN (or @PRT IMSL*DOC.ZXMIN) | Access: LIB NBS*IMSL

ZXMWD Global minimum (with constraints) of a function of n variables. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G2h1a1 | Usage: CALL ZXMWD(FCN,N,NSIG,A,B,NSRCH,X,F,WORK,IWORK,IER) | On-line doc: CALL GAMSDOC ZXMWD (or @PRT IMSL*DOC.ZXMWD) | Access: LIB NBS*IMSL

ZXSSQ Minimum of the sum of squares of m functions in n variables using a finite difference Levenberg-Marquardt algorithm. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): K1b1a1 L8g1a L8g2a | Usage: CALL ZXSSQ (FUNC,M,N,NSIG,EPS,DELTA,MAXFN,IOPT,PARM,X,SSQ,F, XJAC,IXJAC,XJTJ,WORK INFER,IER) | On-line doc: CALL GAMSDOC ZXSSQ (or @PRT IMSL*DOC.ZXSSQ) | Access: LIB NBS*IMSL

LIBRARY REFERENCE

The "library" is the largest unit cataloged in GAMS. A library may be a collection of Fortran subprograms, a collection of stand-alone programs, or even a complex interactive system. In this section we present a summary description of each of the libraries currently listed in GAMS, along with information about how to use them at NBS. Libraries are listed in alphabetical order. The following explains the terminology used in the summaries.

LEGEND

Type	indicates how the software is organized. <i>Subprogram library</i> : a collection of subprograms. In order to use the library, one writes a main program, usually in Fortran, which calls the desired subprogram, and then links the necessary components of the pre-compiled library with the user's main program. <i>Stand-alone program library</i> : a collection of stand-alone programs. In order to use the library, one prepares an input file of commands and data, and then executes the desired program. <i>Program library</i> : a collection containing both subprograms and stand-alone programs. <i>Partitioned library</i> : a library which is made up of smaller collections called <i>packages</i> . A package is a collection of modules, e. g., programs or subprograms, for a narrow class of problems and is obtained from a single developer. <i>Interactive program</i> : a stand-alone program which is executed interactively. The word <i>system</i> may be used when the library corresponds to a single interactive program which performs a large variety of tasks.
Version	indicates the version name, number, or date of the software documented in GAMS. Note that different versions may be implemented on different computers at NBS.
Portability	indicates restrictions on library usage and the ease of transporting the software to other machines. <i>Portable</i> : in the public domain, written in a commonly available subset of the programming language (usually ANSI Standard Fortran), and free of dependencies on the arithmetic properties of a specific machine. The source code for this software is available and may easily be transported to a large number of computers. <i>Portable, some conversion required</i> : while most of the code is portable, certain localized parts of the computation use proprietary software or machine-specific constructs. These portions in general will have to be rewritten to implement this software on different systems. <i>Proprietary</i> : use of the software is governed by a licensing agreement. This license may restrict use to a specific computer or it may allow site-wide use at NBS. Contact a GAMS consultant for further information.
Reference	cites a hard-copy description of the library.
Developer	lists at least one of the institutions responsible for the development of the software, and the name of a contact there.
Support level	indicates the level of support provided for users of the software at NBS. <i>Fully supported</i> : at least one of the developers is on the NBS staff and is available to provide assistance. <i>Supported</i> : help in locating errors in the software is available and liaison with the developers is maintained. <i>Mildly supported</i> : some help in locating errors in the software is available.
Summary doc	indicates how to obtain an on-line summary of the capabilities of the library. For the Sperry 1100, this takes the form "CTS command (EXEC command)".
Detailed doc	indicates how to obtain detailed on-line documentation for individual modules in the library. For the Sperry 1100, this takes the form "CTS command (EXEC command)".
Access	indicates how to gain access to the library. For the Sperry 1100, this takes the form "CTS command (EXEC command)".

BMDP

Type : Stand-alone program library

Version : 1983

Approximately 40 programs for statistical data analysis – data description, line-printer plotting, regression, analysis of variance, frequency tables, time series analysis, multivariate analysis (cluster, correlation, discriminant, factor), and life tables.

Portability : Proprietary

Reference : BMDP Statistical Software, University of California Press, 1983

Developer : BMDP Statistical Software Inc., 1964 Westwood Blvd., Suite 202, Los Angeles, CA 90025

NBS USAGE

On Sperry 1100/FTN ...

Support Level : Supported

Summary Doc : OLD BMDP*DOC.SUMMARY (or @PRT BMDP*DOC.SUMMARY)

Detailed Doc : Refer to BMDP reference manual

Access : PXQT NBS*PLIB\$.BMDP BMDprogram (or @NBS*PLIB\$.BMDP BMDprogram)

On Cyber 750 ...

Support Level : Supported

Summary Doc : GET(BMDPDOC/UN=CAMLIB)
BMDPDOC.

Detailed Doc : Refer to BMDP reference manual

Access : GET(BMDPGET/UN=CAMLIB)
BMDPGET(BMDprogram)
BMDprogram.

Note: The current version of BMDP implemented on the Cyber 750 is the 1982 CDC version.

CONTENTS

Data description	P1D, P2D, P4D, P9D, PAM
Graphics	P5D, P6D
Data manipulation	P1S
Elementary inference	P3D, P3S
Analysis of variance	P7D, P1V-P4V, P8V
Regression	P1R-P6R, P9R, PAR, PLR, P9M
Contingency tables	P4F
Time series analysis	P1T, P2T
Correlation analysis	P8D, P6M
Discriminant analysis	P7M
Factor analysis	P4M, P8M
Cluster analysis	P1M-P3M, PKM, Q3M*
Life testing and survival analysis	P1L, P2L

*Available only in CDC version.

CMLIB

Type : Partitioned Fortran subprogram library

Version : 1983

A collection of high-quality, easily transportable Fortran subroutine packages solving standard problems in many areas of mathematics and statistics.

Portability : Portable

Reference : NBS Core Math Library, vols. 1-4 (Gaithersburg: QA297.c69 in NBS Library, Admin E-120; Boulder: available in Radio Bldg., Rm. 4058)

Developer : Refer to package listings below

NBS USAGE

On Sperry 1100/FTN ...

Support Level : Supported

Summary Doc : CALL GAMSDOC /CMLIB (or @PRT CMLIB*DOC.SUMMARY)

Detailed Doc : CALL GAMSDOC /*package* (or @PRT CMLIB*DOC.SUMMARY/*package*)
CALL GAMSDOC *subprogram* (or @PRT CMLIB*DOC.*subprogram/package*)

Access : ASSUME LIBRARY NBS*CMLIB (or LIB NBS*CMLIB)

Source Availability : Source for each subprogram in a given package is available in the file element
CMLIB*SOURCE.*subprogram/package*

Several CTS command procedures (subroutines) are available for the automatic retrieval of CMLIB source. Procedures with names containing GETMODS retrieve individual subprograms while those with names containing GETPKG retrieve whole packages. Procedures whose names end in X retrieve the requested source plus all external references in CMLIB.

CALL CMLIB*TOOLS.GETMODS *list*
CALL CMLIB*TOOLS.GETMODSX *list*
CALL CMLIB*TOOLS.GETPKG *package*
CALL CMLIB*TOOLS.GETPKGX *package*

Here *list* is a list of subprogram names separated by commas and *package* is the name of a CMLIB package. Output is left in the CTS work area.

On Cyber 750/FTN4 ...

Support Level : Supported

Summary Doc : GET(CAMGUID/UN=CAMLIB)
CAMGUID.

Detailed Doc : See reference above

Access : ATTACH(CMLIB4/UN=CAMLIB,NA)
LIBRARY(CMLIB4)

Source Availability : Source for a given subprogram may be obtained using the commands
GET(CMSOURC/UN=CAMLIB)
CMSOURC(*subprogram*)

On Cyber 750/FTN5 ...

Support Level : Supported

Summary Doc : GET(CAMGUID/UN=CAMLIB)
CAMGUID.

Detailed Doc : See reference above

Access : ATTACH(CMLIB5/UN=CAMLIB,NA)
LIBRARY(CMLIB5)

Source Availability : See description of Cyber 750/FTN4 above.

On Perkin-Elmer 32 bit systems ...

Support Level : Supported

Summary Doc : See reference above

Detailed Doc : See reference above

Access : CMLIB.OBJ

Note: CMLIB is not automatically installed on all PE systems. However, it is available from the Scientific Computing Division upon request. (Contact Richard Freemire at FTS 921-2562.)

CONTENTS

- AMOSLIB** A collection of special function routines with particular emphasis on the special functions of statistics. | Version: 1980 | Reference: D. Amos, Sandia Laboratories Report SAND 77-1390, 1977 | Developer: Sandia Laboratories, Albuquerque, NM 87185 (D. Amos)
- BLAS** Basic linear algebra subroutines. Perform various elementary matrix and vector operations. | Version: 1979 | Reference: ACM Transactions on Mathematical Software, vol. 5 (1979), pp. 308-323 | Developer: Jet Propulsion Laboratory, Pasadena, CA 91103 (C.L. Lawson)
- BSPLINE** Subroutines for computing with piecewise polynomials (B-splines). Includes interpolation, differentiation and integration with B-splines. | Version: 1980 | Reference: C. de Boor, A Practical Guide to Splines, Springer-Verlag, 1978. | Developer: Sandia Laboratories, Albuquerque, NM 87185 (D. Amos)
- BVSUP** Solves systems of linear two-point boundary value problems. | Version: 1982 | Reference: SIAM J. Numerical Analysis, vol. 14 (1977), pp. 40-70 | Developer: Sandia National Laboratories, Albuquerque, NM (M. Scott, H. A. Watts)
- CDRIV** Solves initial value problems for systems of ordinary differential equations including stiff equations. Complex version of SDRIV—solves complex systems with real independent variable. | Version: 1981 | Reference: None | Developer: National Bureau of Standards, Washington, DC 20234 (D. Kahaner), Los Alamos National Laboratory, Los Alamos, NM 87545 (D. Sutherland)
- CPQR79** Finds all zeros of real and complex polynomials via eigenvalue methods. | Version: 1980 | Reference: None | Developer: Sandia Laboratories, Albuquerque, NM 87185 (W. Vandevender)
- CPZERO** Computes all the zeros of real or complex coefficient polynomials. Error bounds are also computed. | Version: 1981 | Reference: None | Developer: National Bureau of Standards, Washington, DC 20234 (D. Kahaner)
- DBSPLINE** Subroutines for computing with piecewise polynomials (B-splines). Double precision version of BSPLINE package. | Version: 1980 | Reference: C. de Boor, A Practical Guide to Splines, Springer-Verlag, 1978. | Developer: Sandia Laboratories, Albuquerque, NM 87185 (D. Amos)
- DDASSL** Solves the system of differential/algebraic equations of the form $g(t, y, y') = 0$, with given initial values. Double precision version of SDASSL. | Version: 1983 | Reference: SIAM J. Scientific and Statistical Computing, vol. 3 (1982), no. 2, pp. 367-384 | Developer: Sandia National Laboratories, Livermore, CA 94550 (L. Petzold)
- DDRIV** Solves initial value problems for systems of ordinary differential equations, including stiff systems. (Double precision version of SDRIV). | Version: 1980 | Reference: None | Developer: National Bureau of Standards, Washington, DC 20234 (D. Kahaner), Los Alamos National Laboratory, Los Alamos, NM 87545 (D. Sutherland)
- DEPAC** Solves systems of first order ordinary differential equations with arbitrary initial data. | Version: 1982 | Reference: None | Developer: Sandia National Laboratories, Albuquerque, NM 87185 (L. Shampine, H. A. Watts)
- DNL2SN** Solves nonlinear least squares problems and general optimization problems. (Double precision version of NL2SN.). | Version: 1982 | Reference: ACM Transactions on Mathematical Software, vol. 7 (1981),

pp. 369-383, and MIT/CCREMS Report TR-18 | Developer: Massachusetts Institute of Technology, Cambridge, MA (D. Gay)

- DQRLSS** Solves linear least squares problems in matrix form $Ax=b$. Easy-to-use driver for LINPACK routines. Double precision version of SQRLSS. | Version: 1983 | Reference: J.J. Dongarra, et.al., LINPACK User's Guide, SIAM, Philadelphia, 1979 (QA214.L56 in NBS Library, Admin E-120, Gaithersburg) | Developer: National Bureau of Standards, Washington, DC 20234 (D. Kahaner)
- DTENSORBS** Interpolation of gridded data in two or three dimensions using tensor products of one-dimensional B-splines. Double precision version of TENSORBS. | Version: 1982 | Reference: C. de Boor, A Practical Guide to Splines, Springer-Verlag, 1978 | Developer: National Bureau of Standards, Washington, DC 20234 (R. Boisvert)
- EISPACK** Solves various linear algebraic eigenvalue problems. | Version: 1977 | Reference: B.T. Smith et. al., Matrix Eigensystem Routines, EISPACK Guide, Springer, 1976. | Developer: Argonne National Laboratory, Argonne, IL 60439 (B. Garbow)
- FC** Solves constrained least squares problems. | Version: 1980 | Reference: R.J. Hanson, Sandia Laboratories Report SAND 78-1291, 1978 | Developer: Sandia Laboratories, Albuquerque, NM 87185 (R. Hanson, K. Haskell)
- FCNPAK** Subprograms to compute various special functions not readily available elsewhere. At present the package contains subroutines for the associated Legendre (Ferrers) functions and the normalized Legendre polynomials. | Version: 1982 | Reference: ACM Transactions on Mathematical Software, vol. 7 (1981), pp. 93-105 and 141-146 | Developer: National Bureau of Standards, Washington, DC 20234 (D. W. Lozier, J. M. Smith)
- FFTPKG** Subroutines for computing the fast Fourier transform in various forms. | Version: 1980 | Reference: "Efficient Subprograms for the Solution of Elliptic Partial Differential Equations," TN/IA-109, NCAR, Boulder, CO 80307 | Developer: National Center for Atmospheric Research, Boulder, CO 80307 (P. Swarztrauber)
- FNLIB** Portable special function routines. | Version: 1979 | Reference: L.W. Fullerton, in Portability of Numerical Software, Springer, 1977, pp. 452-83 | Developer: Bell Laboratories, Murray Hill, NJ 07974 (L.W. Fullerton)
- FSHPK** Solves separable elliptic boundary value problems in two and three dimensions in a variety of coordinate systems. | Version: 3.1 | Reference: "Efficient Subprograms for the Solution of Elliptic Partial Differential Equations," TN/IA-109, NCAR, Boulder, CO 80307 | Developer: National Center for Atmospheric Research, Boulder CO (P.Swarztrauber), National Bureau of Standards, Boulder, CO 80303 (R. Sweet)
- LICEPACK** Solves linear algebraic eigenvalue problems. (Provides an interface to the EISPACK package). | Version: 1980 | Reference: None | Developer: National Bureau of Standards, Washington, DC 20234 (D. Kahaner)
- LINDRIVES** Programs to solve various types of linear systems of algebraic equations. Provides an easy to use interface to the LINPACK package. | Version: 1980 | Reference: None | Developer: Los Alamos Scientific Laboratory, Los Alamos NM 87545
- LINPACKC** Analyse and solve various systems of linear algebraic equations. (Complex precision version of LINPACK). | Version: 1978 | Reference: J.J. Dongarra et. al., LINPACK User's Guide, SIAM, Philadelphia, 1979 (QA214.L56 in NBS Library, Admin E-120, Gaithersburg) | Developer: Argonne National Laboratory, Argonne, IL 60439 (J.J. Dongarra)
- LINPACKD** Analyse and solve various systems of linear algebraic equations. (Double precision version of LINPACK). | Version: 1978 | Reference: J.J. Dongarra, et.al., LINPACK User's Guide, SIAM, Philadelphia, 1979 (QA214.L56 in NBS Library, Admin E-120, Gaithersburg) | Developer: Argonne National Laboratory, Argonne, IL 60439 (J.J. Dongarra)
- LINPACKS** Analyse and solve various systems of linear algebraic equations. (Single precision version of LINPACK). | Version: 1978 | Reference: J.J. Dongarra, et.al., LINPACK User's Guide, SIAM, Philadelphia, 1979 (QA214.L56 in NBS Library, Admin E-120, Gaithersburg) | Developer: Argonne National Laboratory, Argonne, IL 60439 (J.J. Dongarra)
- LOTPS** Produces a smooth interpolant to "scattered" e.g. non-gridded, data in the plane. | Version: 1982 | Reference: R. Franke, Naval Postgraduate School Tech. Reports | Developer: Naval Postgraduate School, Monterey, CA (R. Franke)

- MACHCONST** Functions that return machine-dependent constants. | Version: 1979 | Reference: ACM transactions on Mathematical Software, vol. 4 (1978), pp. 177-188 | Developer: Bell Laboratories, Murray Hill, NJ 07974 (P. Fox)
- MAXENTROPY** Subprograms for computing maximum entropy spectrum estimates for equally spaced time series data. | Version: 1983 | Reference: None | Developer: US Naval Underwater Systems Center, Newport, RI (Manuel T. Silva)
- NL2SN** Solves nonlinear least squares problems and general nonlinear unconstrained optimization problems. | Version: 1982 | Reference: ACM Transactions on Mathematical Software, vol. 7 (1981), pp. 369-383, and MIT/CCREMS Report TR-18 | Developer: Massachusetts Institute of Technology, Cambridge, MA (D. M. Gay)
- PCHIP** Produces aesthetic looking interpolants to univariate data by using piecewise cubic Hermite functions. | Version: 1982 | Reference: F. Fritsch, Lawrence Livermore Laboratory Report UCRL-85104 | Developer: Lawrence Livermore Laboratory, Livermore, CA 94550 (F. Fritsch)
- Q1DA** Automatic evaluation of one-dimensional integrals of a user-defined function of one variable. | Version: 1982 | Reference: None | Developer: National Bureau of Standards, Washington, DC 20234 (D. Kahaner)
- QUADDP** Subprograms for evaluating definite integrals of functions of one variable; double precision version of QUADSP. | Version: 1981 | Reference: R. Piessens, et.al., QUADPACK, A Subroutine Package for Automatic Integration, Springer-Verlag, 1983 | Developer: Univ. of Leuven, Heverlee, Belgium (R. Piessens), Western Michigan Univ., Kalamazoo, MI 49008 (E. de Donker), National Bureau of Standards, Washington, DC 20234 (D. Kahaner)
- QUADSP** Subprograms for evaluating definite integrals of functions of one variable; including singular integrands and infinite intervals. | Version: 1981 | Reference: R. Piessens, et.al., QUADPACK, A Subroutine Package for Automatic Integration, Springer-Verlag, 1983 | Developer: Univ. Of Leuven, Heverlee, Belgium (R. Piessens), Western Michigan Univ., Kalamazoo, MI 49008 (E. de Donker) National Bureau of Standards, Washington, DC 20234 (D. Kahaner)
- RV** A collection of portable pseudo-random number generators. | Version: 1983 | Reference: None | Developer: National Bureau of Standards, Washington, DC 20234 (D. Kahaner, J. Blue) Univ. of Washington, Seattle, WA (G. Marsaglia)
- SDASSL** Solves the system of differential/algebraic equations of the form $g(t, y, y') = 0$, with given initial values. | Version: 1983 | Reference: SIAM J. Scientific and Statistical Computing, vol. 3, no. 2 (1982), pp. 367-384. | Developer: Sandia National Laboratories, Livermore, CA 94550 (L. Petzold)
- SDRIV** Solves initial value problems for systems of ordinary differential equations, including stiff systems. | Version: 1980 | Reference: None | Developer: National Bureau of Standards, Washington, DC 20234 (D. Kahaner), Los Alamos National Laboratory, Los Alamos, NM 87545 (D. Sutherland)
- SGLSS** Solves $AX=B$ for either overdetermined systems (least squares problems) or underdetermined systems (least length problems) with reliable rank deficiency determination. Uses Householder transformations. | Version: 1983 | Reference: T. Manteuffel, Sandia Laboratory Report SAND80-0655, June 1980 | Developer: Los Alamos National Laboratory, Los Alamos, NM 87545 (T. Manteuffel)
- SLRPACK** A collection of subprograms for simple linear regression. | Version: 1981 | Reference: None | Developer: National Bureau of Standards, Washington, DC 20234 (S. Howe)
- SLVBLK** Solves linear systems of algebraic equations where the coefficient matrix is in "almost block diagonal" form. | Version: 1980 | Reference: ACM Transactions on Mathematical Software, vol. 6 (1980), pp. 80-91 | Developer: Mathematics Research Center, Univ. of Wisconsin, Madison, WI (C. deBoor, R. Weiss)
- SNLS1E** A suite of codes for nonlinear least squares problems (and computation of associated covariance matrices) and systems of nonlinear equations. | Version: 1983 | Reference: None | Developer: Sandia National Laboratory, Albuquerque, NM 87185 (K. Hiebert)
- SPLP** Solves linear optimization problems. | Version: 1982 | Reference: None | Developer: Sandia National Laboratories, Albuquerque, NM 87185 (R. L. Hanson)
- SQRLSS** Solves linear least square problems in the matrix form $Ax=b$. Easy-to-use driver for LINPACK routines. | Version: 1981 | Reference: None | Developer: National Bureau of Standards, Washington, DC 20234 (D. Kahaner), Univ. of New Mexico, Albuquerque, NM (C. Moler)
- SSORT** Fast in-core sorting of arrays. | Version: 1979 | Reference: None | Developer: Sandia Laboratories,

Albuquerque, NM 87185 (R. E. Jones), National Bureau of Standards, Washington, DC 20234 (D. Kahaner)

- SUDSSODS** Solves overdetermined and underdetermined systems of linear equations in the least squares sense. | Version: 1981 | Reference: H.A. Watts, Sandia Laboratories Report SAND77-0683. | Developer: Sandia National Laboratories, Albuquerque, NM 87185 (H. A. Watts)
- TENSORBS** Interpolation of gridded data in two or three dimensions using tensor products of one-dimensional B-splines. | Version: 1982 | Reference: C. de Boor, A Practical Guide to Splines, Springer-Verlag, 1978 | Developer: National Bureau of Standards, Washington, DC 20234 (R. Boisvert)
- XBLAS** Extended basic linear algebra subroutines. Perform various matrix and vector operations not found in the BLAS. | Version: 1980 | Reference: None | Developer: National Bureau of Standards, Washington, DC 20234 (D. Kahaner)
- XERROR** Error handling utilities. | Version: 1980 | Reference: None | Developer: Sandia Laboratories, Albuquerque, NM 87185 (R.E. Jones)
- YSMP** The Yale Sparse Matrix Package. Solves large sparse systems of linear algebraic equations in core with no pivoting. | Version: 1980 | Reference: S.C. Eisenstat et. al., The Yale Sparse Matrix Package, Yale C.S. Dept. Report. | Developer: Dept. of Computer Science, Yale Univ., New Haven, CT 06520 (S. Eisenstat)
- ZEROIN** Finds zeros of a function of one variable. | Version: 1979 | Reference: L. Shampine et. al., Numerical Computing: An Introduction, Saunders, 1973 | Developer: Sandia Laboratories, Albuquerque, NM 87185 (L.F. Shampine)

DATAPAC

Type : Fortran subprogram library

Version : 77.5

Approximately 170 subroutines for probability distribution, density, percent point, and sparsity function evaluation, random number generation, line-printer plotting, histograms, scatter diagrams, probability plots, data manipulation, general statistical analysis, time series analysis, polynomial regression, and ANOVA.

Portability : Portable
Reference : J. J. Filliben, User's Guide to Datapac (version 77.5)
Developer : National Bureau of Standards, Washington, DC 20234 (J. J. Filliben)

NBS USAGE

On Sperry 1100/FTN ...

Support Level : Fully supported
Summary Doc : CALL GAMSDOC /DATAPAC (or @PRT DATAPAC*DOC.SUMMARY)
Detailed Doc : CALL GAMSDOC *subprogram* (or @PRT DATAPAC*DOC.*subprogram*)
Access : ASSUME LIBRARY NBS*DAPAC (or LIB NBS*DAPAC)
Source Availability : Source for a given subprogram is available in the file element
 DATAPAC*SOURCE.*subprogram*

On Cyber 750/FTN4 ...

Support Level : Fully supported
Summary Doc : GET(CAMGUID/UN=CAMLIB)
 CAMGUID.
Detailed Doc : See reference above (or refer to source code)
Access : ATTACH(DATAPAC/UN=CAMLIB,NA)
 LIBRARY(DATAPAC)
Source Availability: Source for a given subprogram may be obtained using the commands
 ATTACH(DPACMOD/UN=CAMLIB,NA)
 MODIFY(P=DPACMOD,Z,LO=E)/ *DECK *subprogram*

CONTENTS

Individual statistics	(20 subprograms)
Data manipulation	(17 subprograms)
Probability plots	(19 subprograms)
Printer (wide-carriage) plots	(12 subprograms)
Terminal (narrow-width) plots	(8 subprograms)
General analyses	(10 subprograms)
Cumulative distribution function values	(21 subprograms)
Probability density function values	(7 subprograms)
Percent point function values	(20 subprograms)
Sparsity function values	(7 subprograms)
Random number generators	(23 subprograms)
Polynomial regression	(4 subprograms)
Time series analyses	(5 subprograms)
Input/output	(4 subprograms)

IMSL

Type : Fortran subprogram library

Version : Edition 9

Approximately 500 Fortran subprograms solving standard problems in many areas of mathematics and statistics.

Portability : Proprietary
Reference : IMSL Reference Manual, vols. 1-4 (Gaithersburg: QA297.i19 in NBS Library, Admin E-120;
Boulder: QA276.25.i58 in Radio Bldg. Library)
Developer : IMSL Inc., 6th Floor NBC Bldg., 7500 Bellaire Blvd., Houston, TX 77036

NBS USAGE

On Sperry 1100/FTN ...

Support Level : Supported
Summary Doc : See reference above
Detailed Doc : CALL GAMSDOC *subprogram* (or @PRT IMSL*DOC.*subprogram*)
Access : ASSUME LIBRARY NBS*CMLIB (or LIB NBS*IMSL)

On Cyber 750/FTN4 ...

Support Level : Supported (by NOAA/SEL)
Summary Doc : GET(LIBDOC/UN=LIBDOC)
LIBDOC(IMSL)
Detailed Doc : See reference above
Access : ATTACH(IMSL/UN=LIB,NA)
LIBRARY(IMSL)

On Cyber 750/FTN5 ...

Support Level : Supported (by NOAA/SEL)
Summary Doc : GET(LIBDOC5/UN=LIBDOC5)
LIBDOC5(IMSL)
Detailed Doc : See reference above
Access : ATTACH(IMSL/UN=LIB5,NA)
LIBRARY(IMSL)

CONTENTS

Chapter A : Analysis of Variance	(17 subprograms)
Chapter B : Basic Statistics	(22 subprograms)
Chapter C : Categorized Data Analysis	(5 subprograms)
Chapter D : Differential Equations, Quadrature, Differentiation	(14 subprograms)
Chapter E : Eigensystem Analysis	(11 subprograms)
Chapter F : Forecasting, Econometrics, Time series, Transforms	(23 subprograms)
Chapter G : Generation and Testing of Random Numbers	(38 subprograms)
Chapter I : Interpolation, Approximation, Smoothing	(16 subprograms)
Chapter L : Linear Algebraic Equations	(42 subprograms)
Chapter M : Mathematical and Statistical Special Functions	(67 subprograms)
Chapter N : Non-parametric Statistics	(25 subprograms)
Chapter O : Observation Structure, Multivariate Statistics	(18 subprograms)
Chapter R : Regression Analysis	(29 subprograms)
Chapter S : Sampling	(8 subprograms)

Chapter U : Utility Functions	(28 subprograms)
Chapter V : Vector-Matrix Arithmetic	(72 subprograms)
Chapter Z : Zeros and Extrema, Linear Programming	(23 subprograms)

NOTES

- (a) The IMSL library is available on a large number of computer systems, and the precision of each routine may vary from implementation to implementation. For example, some routines may be only available in single precision on 60-bit machines while only in double precision on 32-bit machines. Other routines may not be available on all machines. Users should consult the IMSL Library Reference Manual to determine the actual precision of each routine in each implementation.

INVAR

Type : Stand-alone interactive program library

Version : 1983

Interactive programs for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results and DISSPLA* graphics.

Portability : Portable, some conversion required*
Reference : C. Wolfe and B. W. Rust, NBS Tech. Note, in preparation
Developer : National Bureau of Standards, Washington, DC 20234 (B. Rust)

NBS USAGE

On Sperry 1100 ...

Support Level : Fully supported
Summary Doc : CALL GAMSDOC /INVAR (or @PRT INVAR*DOC.SUMMARY)
Detailed Doc : See summary documentation.
Access : CALL SCD*CTSLIB.*program* (in CTS only)

CONTENTS

INVAR1 version with line-printer graphics only.
INVAR2 version with DISSPLA* graphics.

NOTES

*Graphics are implemented via calls to the DISSPLA proprietary graphics subprogram library (ISSCO, 4186 Sorrento Valley Blvd., San Diego, CA 92121). The remainder of the program is in the public domain.

MATHWARE

Type : Partitioned program library

Version : 1980

A collection of packages, subroutines and other processors useful in solving a variety of scientific computing problems.

Portability : Portable, some conversion required

Reference : See package listings below

Developer : See package listings below

NBS USAGE

On Sperry 1100 ...

Support Level : Mildly supported

Summary Doc : OLD MATHWARE*DOC.SUMMARY (or @PRT MATHWARE*DOC.SUMMARY)

Detailed Doc : Refer to file MATHWARE**package*.

Access : See individual sublibrary documentation

CONTENTS

- ITPACK** Seven programs for the solution to large sparse systems of linear equations using adaptive accelerated iterative methods. (Best suited for symmetric positive definite systems.). | Version: 2A | Reference: ACM Transactions on Mathematical Software, vol. 8 (1982), pp. 302-322 | Developer: Center for Numerical Analysis, Univ. of Texas, Austin, TX 78712 (D. Kincaid, D. Young)
- NASHLIB** Programs for computations in linear algebra and function minimization specially designed for use on small computers. | Version: 1980 | Reference: J.C. Nash, Compact Numerical Methods for Computers, Halsted Press, 1980 | Developer: Univ. of Ottawa, Ottawa, Ont., Canada (J.C. Nash)
- OLIVER** Computes 1st, 2nd or 3rd derivative of a given user defined function $f(x)$. | Version: 1980 | Reference: Algorithm 17, J. Computational and Applied Mathematics, vol. 6, no. 2, 1980 | Developer: Dept of Computer Sci., Univ. of Essex, Colchester, Essex, England (J. Oliver)
- STEGUN** A collection of Fortran subprograms for computing various special functions. | Version: 1980 | Reference: None | Developer: National Bureau of Standards, Washington, DC 20234 (I. Stegun)

MATLAB

Type : Interactive system

Version : 1983

A system for matrix calculations, including solving linear systems, linear least squares problems, eigenvalue and eigenvector calculations, QR decomposition, singular value decomposition, and inverses. Based on LINPACK and EISPACK software.

Portability : Portable, some conversion required
Reference : C. Moler, MATLAB User's Guide, Univ. of NM Tech. Rept. CS 81-1
Developer : University of New Mexico, Albuquerque, NM (C. Moler)

NBS USAGE

On Sperry 1100 ...

Support Level : Supported
Summary Doc : OLD MATLAB*DOC.SUMMARY (or @PRT MATLAB*DOC.SUMMARY)
Detailed Doc : HELP *command* (in MATLAB)
Access : XQT NBS*MATLAB.MATLAB (or @XQT NBS*MATLAB.MATLAB)

On Perkin-Elmer 32 bit systems ...

Support Level : Supported
Summary Doc : HELP (in MATLAB)
Detailed Doc : HELP *command* (in MATLAB)
Access : MATLAB

Note: MATLAB is not automatically installed on all PE systems. However, it is available from the Scientific Computing Division upon request. (Contact Richard Freemire at FTS 921-2562.)

MINITAB

Type : Interactive system

Version : 1982

Approximately 150 commands for data manipulation, plotting, random number generation, general purpose statistical analysis including regression, time series, EDA (exploratory data analysis), ANOVA, and analysis of tables.

Portability : Proprietary

Reference : T.A. Ryan, Jr., B.L. Joiner, and B.F. Ryan, Minitab Reference Manual, 1982

Developer : Minitab Project, 215 Pond Laboratory, University Park, PA 16802 (T. Ryan)

NBS USAGE

On Sperry 1100 ...

Support Level : Supported

Summary Doc : OLD MINITAB*DOC.SUMMARY (or @PRT MINITAB*DOC.SUMMARY)

Detailed Doc : HELP *command* (in Minitab)

Access : CALL MINITAB (or @XQT NBS*MINITAB.MINITAB)

On Perkin-Elmer 32 bit systems ...

Support Level : Supported

Summary Doc : HELP (in Minitab)

Detailed Doc : HELP *command* (in Minitab)

Access : MINITAB

Note: Minitab is not automatically installed on all PE systems. However, it is available from the Scientific Computing Division upon request. (Contact Richard Freemire at FTS 921-2562.)

CONTENTS

Column operations and row operations	(21 commands)
Editing	(4 commands)
Manipulation	(5 commands)
Sorting	(3 commands)
Arithmetic	(8 commands)
Functions	(16 commands)
Histograms and scatter diagrams	(6 commands)
Probability functions	(3 commands)
Statistics on one column of data	(7 commands)
Random number generation	(8 commands)
Correlation analysis	(1 command)
Analysis of variance	(7 commands)
Regression	(2 commands)
Tables	(3 commands)
Time series analyses	(6 commands)
EDA (exploratory data analysis)	(9 commands)
Input/output	(14 commands)
Macros and loops	
Help	

NAG

Type : Fortran subprogram library

Version : Mark 10

Approximately 500 Fortran subroutines for solving standard problems in many areas of mathematics, statistics and optimization.

Portability : Proprietary
Reference : NAG Fortran Library Manual, vols. 1-6 (Gaithersburg: QA297.n3 in NBS Library, Admin E-120; Boulder: consult User Services reference collections)
Developer : Numerical Algorithms Group Ltd., 7 Banbury Rd., Oxford OX2 6NN, England (or NAG Inc., 1131 Warren Ave., Downers Grove, IL 60515)

NBS USAGE

On Sperry 1100/FTN ...

Support Level : Supported
Summary Doc : CALL GAMSDOC /NAG (or @PRT NAG*DOC.SUMMARY)
Detailed Doc : CALL GAMSDOC *subprogram* (or @PRT NAG*DOC.*subprogram*)
Access : ASSUME LIBRARY NBS*NAG (or LIB NBS*NAG)

Note: in the Sperry implementation all NAG subprograms are available in both single and double precision versions. Single precision routines have names which end in the letter E, while double precision routines have names which end in the letter F.

On Cyber 750/FTN4 ...

Support Level : Supported
Summary Doc : GET(NAGSUM/UN=CAMLIB)
COPY(NAGSUM)
Detailed Doc : See reference above
Access : ATTACH(NAGLIB4/UN=CAMLIB,NA)
LIBRARY(NAGLIB4)

Note: Only routines ending whose names end in F are available in the CDC implementation of the NAG library. These programs are *single* precision.

On Cyber 750/FTN5 ...

Support Level : Supported
Summary Doc : GET(NAGSUM/UN=CAMLIB)
COPY(NAGSUM)
Detailed Doc : See reference above
Access : ATTACH(NAGLIB5/UN=CAMLIB,NA)
LIBRARY(NAGLIB5)

Note: Only routines ending whose names end in F are available in the CDC implementation of the NAG library. These programs are *single* precision.

CONTENTS

Chapter A02 : Complex Arithmetic	(3 subprograms)
Chapter C02 : Zeros of Polynomials	(2 subprograms)
Chapter C05 : Roots of One or More Transcendental Equations	(11 subprograms)

Chapter C06 : Summation of Series	(12 subprograms)
Chapter D01 : Quadrature	(25 subprograms)
Chapter D02 : Ordinary Differential Equations	(34 subprograms)
Chapter D03 : Partial Differential Equations	(9 subprograms)
Chapter D04 : Numerical Differentiation	(1 subprograms)
Chapter D05 : Integral Equations	(2 subprograms)
Chapter E01 : Interpolation	(7 subprograms)
Chapter E02 : Curve and Surface Fitting	(22 subprograms)
Chapter E04 : Maximizing or Minimizing a Function	(35 subprograms)
Chapter F01 : Matrix Operations, including Inversion	(60 subprograms)
Chapter F02 : Eigenvalue and Eigenvectors	(32 subprograms)
Chapter F03 : Determinants	(9 subprograms)
Chapter F04 : Simultaneous Linear Equations	(26 subprograms)
Chapter F05 : Orthogonalization	(2 subprograms)
Chapter G01 : Simple Calculations on Statistical Data	(17 subprograms)
Chapter G02 : Correlation and Regression Analysis	(26 subprograms)
Chapter G04 : Analysis of Variance	(4 subprograms)
Chapter G05 : Random Number Generators	(32 subprograms)
Chapter G08 : Nonparametric Statistics	(9 subprograms)
Chapter G13 : Time Series Analysis	(18 subprograms)
Chapter H : Operations Research	(7 subprograms)
Chapter M01 : Sorting	(20 subprograms)
Chapter P01 : Error Trapping	(1 subprograms)
Chapter S : Approximations of Special Functions	(41 subprograms)
Chapter X01 : Mathematical Constants	(2 subprograms)
Chapter X02 : Machine Constants	(15 subprograms)
Chapter X03 : Innerproducts	(2 subprograms)
Chapter X04 : Input/Output Utilities	(2 subprograms)

NOTES

- (a) THE NAG library is available for a wide variety of machines/compiler. "Standard precision" routines in each implementation end in the letter F (for Fortran). On some machines this may represent a single precision routine while on others it may represent a double precision routine. In some implementations both precisions are available. In this case the alternate precision routines end in E if they are single precision or D if they are double precision. Users are urged to consult the NAG Fortran Library Manual to determine which routines are available in the version implemented for their machine.

PDELIB

Type : Fortran subprogram library

Version : 1982

An informal collection of portable, public-domain Fortran subprograms which solve general systems of nonlinear initial-boundary-value partial differential equations in one or two space dimensions. Each program is based upon the method of lines.

Portability : Portable
Reference : See subprogram listing below.
Developer : See subprogram listing below.

NBS USAGE

On Sperry 1100/FTN ...

Support Level : Supported
Summary Doc : CALL GAMSDOC /PDELIB (or @PRT PDELIB*DOC.SUMMARY)
Detailed Doc : CALL GAMSDOC *subprogram* (or @PRT PDELIB*DOC.*subprogram*)
Access : ASSUME LIBRARY NBS*PDELIB (or LIB NBS*PDELIB)

CONTENTS

MOL1D Solves systems of linear or nonlinear initial-boundary-value problems in one space dimension. Can solve hyperbolic equations with or without discontinuities, parabolic equations (including reaction-diffusion equations). Uses the method of lines based on equi-spaced finite differences. Graphical output available. | Author: J.M. Hyman, Los Alamos National Lab, Los Alamos, NM 87545

PDECOL Solves general nonlinear systems of initial-boundary-value problems in one space dimension with general boundary conditions. Spatial derivatives may be of at most second order. Uses method of lines based on collocation of B-spline basis functions. | Reference: ACM Transactions on Mathematical Software, vol. 5 (1979), pp. 326-351 | Authors: N. Madsen, Lawrence Livermore National Lab, Livermore, CA 94550 and R. Sincovec, Univ. of Colorado at Colorado Springs, Colorado Springs, CO 80907

PDETWO Solves general nonlinear systems of initial-boundary-value problems in two spatial dimensions with quasi-linear boundary conditions. Uses the method of lines based upon finite differences on a user-specified rectangular mesh. | Reference: ACM Transactions on Mathematical Software, vol. 7 (1981), pp. 126-135 | Authors: D. Melgaard, J&M Systems Consultants Ltd., 2430 San Mateo N.E., Albuquerque, NM 87110 and R. Sincovec, Univ. of Colorado at Colorado Springs, Colorado Springs, CO 80907

PLOD

Type : Stand-alone interactive program

Version : 1983

Solves systems of ordinary differential equations. User may change various conditions, parameters, intervals, etc., interactively and plot results on an Hewlett Packard or Tektronix terminal. Very easy to use. Almost no programming experience required.

Portability : Portable, some conversion required*

Reference : None

Developer : National Bureau of Standards, Washington, DC 20234 (D. Kahaner)

NBS USAGE

On Sperry 1100/FTN ...

Support Level : Fully supported

Summary Doc : OLD PLOD*DOC.SUMMARY (or @PRT PLOD*DOC.SUMMARY)

Detailed Doc : See summary documentation

Access : XQT NBS*PLOD.PLOD (or @XQT NBS*PLOD.PLOD)

NOTES

*Graphics are implemented via calls to the DISSPLA proprietary graphics subprogram library (ISSCO, 4186 Sorrento Valley Blvd., San Diego, CA 92121). The remainder of the program is in the public domain.

PORT

Type : Fortran subprogram library

Version : 2

Approximately 200 Fortran subroutines for problems in areas such as curve fitting, ordinary differential equations, integration, etc.

Portability : Proprietary

Reference : The PORT Mathematical Subroutine Library, Manual (QA297.P2 in NBS Library, Admin E-120, Gaithersburg)

Developer : Bell Laboratories, 600 Mountain Ave., Murray Hill, NJ 07974 (P. Fox)

NBS USAGE

On Sperry 1100/FTN ...

Support Level : Supported

Summary Doc : CALL GAMSDOC /PORT

Detailed Doc : CALL GAMSDOC *subprogram* (or @PRT PORT*DOC.*subprogram*)

Access : ASSUME LIBRARY NBS*PORT (or LIB NBS*PORT)

CONTENTS

Approximation	(35 subprograms)
Arithmetic	(8 subprograms)
Differential Equations	(4 subprograms)
Linear Algebra	(6 subprograms)
Optimization	(1 subprogram)
Probability-Statistics	(3 subprograms)
Quadrature and Differentiation	(9 subprograms)
Roots	(5 subprograms)
Special Functions	(18 subprograms)
Transforms	(6 subprograms)
Utility	(88 subprograms)

SLDGL

Type : Fortran subprogram library

Version : 1.1

Fortran subprograms for the numerical solution of initial value problems and boundary value problems for ordinary differential equations, and for elliptic and parabolic partial differential equations.

Portability : Proprietary

Reference : Advances in Computer Methods for Partial Differential Equations, vol. IV (1981), IMACS, pp. 117-125.

Developer : Universitat Karlsruhe, D-7500 Karlsruhe, W. Germany (W. Schonauer)

NBS USAGE

On Sperry 1100/FTN ...

Support Level : Supported

Summary Doc : CALL GAMSDOC /SLDGL (or @PRT SLDGL*DOC.SUMMARY)

Detailed Doc : CALL GAMSDOC *subprogram* (or @PRT SLDGL*DOC.*subprogram*)

Access : ASSUME LIBRARY NBS*SLDGL (or LIB NBS*SLDGL)

CONTENTS

Ordinary initial value problems	(4 subprograms)
Ordinary boundary value problems	(5 subprograms)
Two-dimensional elliptic boundary value problems on rectangular domains	(6 subprograms)
Two-dimensional elliptic boundary value problems on non-rectangular domains	(1 subprogram)
Three-dimensional elliptic boundary value problems on rectangular boxes	(1 subprogram)
Initial boundary value problems for parabolic equations in one space dimension	(8 subprograms)
Initial boundary value problems for parabolic equations in two space dimensions	(4 subprograms)
Initial boundary value problems for parabolic equations in three space dimensions	(1 subprogram)

NOTES

- (a) In each case n nonlinear equations in n unknown functions are admitted.
- (b) When multiple subprograms are available users must choose between types of initial conditions, grids, or numerical methods. Modules which adaptively choose the spatial mesh and the order of accuracy of the spatial discretization scheme are available.
- (c) All subprograms are based upon finite difference methods. Time integration is done using a variable-order variable-step scheme. Solution of discrete elliptic equations is done by either direct (band elimination) or iterative (relaxation) methods.

SPECTRLAN

Type : Stand-alone interactive program

Version : 1983

An interactive spectral analysis program for time series data. Produces periodograms, cumulative periodograms, continuous Fourier power spectra, cumulative power spectra, Fourier amplitude spectra, maximum entropy spectra, and integrated maximum entropy spectra. With DISSPLA* graphics.

Portability : Portable, some conversion required*
Reference : C. Wolfe and B. Rust, NBS Tech. Note, in preparation
Developer : National Bureau of Standards, Washington, DC 20234 (B. Rust)

NBS USAGE

On Sperry 1100/FTN ...

Support Level : Fully supported
Summary Doc : CALL GAMSDOC /SPECTRLAN (or @PRT SPECTRLAN*DOC.SUMMARY)
Detailed Doc : See summary documentation
Access : CALL SCD*CTSLIB.SPECAN (in CTS only)

NOTES

*Graphics are implemented via calls to the DISSPLA proprietary graphics subprogram library (ISSCO, 4186 Sorrento Valley Blvd., San Diego, CA 92121). The remainder of the program is in the public domain.

STATLIB

Type : Fortran subprogram library

Version : 1978

Fifty-four Fortran subroutines for statistical data analysis. The easy-to-use subroutines, which automatically provide comprehensive printing and plotting, do elementary plotting, analyze univariate samples, do linear and nonlinear regression analysis, and time series analysis.

Portability : Portable
Reference : STATLIB: A Library of FORTRAN Subroutines for Statistical Analysis of Experimental Data, 1978 edition
Developer : National Bureau of Standards, Mailcode 714, 325 Broadway, Boulder, CO 80303 (J.R. Donaldson)

NBS USAGE

On Sperry 1100/FTN ...

Support Level : Fully supported
Summary Doc : CALL GAMSDOC /STATLIB (or @PRT STATLIB*DOC.SUMMARY)
Detailed Doc : CALL GAMSDOC *subprogram* (or @PRT STATLIB*DOC.*subprogram*)
Access : ASSUME LIBRARY NBS*STATLIB (or LIB NBS*STATLIB)

On Cyber 750/FTN4 ...

Support Level : Fully Supported
Summary Doc : GET(CAMGUID/UN=CAMLIB)
CAMGUID.
Detailed Doc : See reference above
Access : ATTACH(STATLIB/UN=CAMLIB,NA)
LIBRARY(STATLIB)

On Cyber 750/FTN5 ...

Support Level : Fully Supported
Summary Doc : GET(CAMGUID/UN=CAMLIB)
CAMGUID.
Detailed Doc : See reference above
Access : ATTACH(SL2FTN5/UN=CAMLIB,NA)
LIBRARY(SL2FTN5)

CONTENTS

Plotting (histograms, Y vs. X, symbol, multiple, time series)	(21 subprograms)
Analysis of a single random sample	(3 subprograms)
Correlation analysis	(2 subprograms)
Random number generation	(1 subprogram)
One-way analysis of variance	(2 subprograms)
Linear least squares analysis	(8 subprograms)
Nonlinear least squares analysis	(4 subprograms)
Time series analyses	(16 subprograms)

A

Adams' predictor-corrector method (*see* ORDINARY DIFFERENTIAL EQUATIONS)
Adaptive quadrature (*see* QUADRATURE)
Airy function (*see* SPECIAL FUNCTIONS)
Aitken's method (*see* INTERPOLATION)
Algebraic-differential systems (*see* ORDINARY DIFFERENTIAL EQUATIONS)
Analysis of variance and analysis of covariance (*see* EXPERIMENTAL DESIGN)
Anger functions (*see* SPECIAL FUNCTIONS)

APPROXIMATION K

Chebyshev K2
Constrained linear least squares K1a2
Constrained nonlinear least squares K1b2
Curve fitting K1a1a
Evaluation of fitted functions K6
 L_1 K3
 L_2 K1a
 L_∞ K2
Least absolute value (L_1) K3
Levenberg-Marquardt method K1b1a
Linear least square (L_2) K1a
Mesh generation K6
Minimax (Chebyshev or L_∞) K2
Multivariate K1a1b
Nonlinear least squares K1b
Pade K4
Piecewise polynomials K1a1a1
Polynomial splines K1a1a1
Polynomials K1a1a2
Rational functions K1a1a3
Service routines (e.g., mesh generation, evaluation of fitted functions) K6
Smoothing K5
Splines K1a1a1, K1a1b, K5, K6
Surface fitting K1a1b
Taylor polynomial K4
Trigonometric functions K1a1a3
Unconstrained linear least squares K1a1
Unconstrained nonlinear least squares K1b1

also see REGRESSION

ARITHMETIC A

Base conversion A6b
Change of representation A6
Complex (single, double, and extended precision, extended range) A4
Construction of machine-base numbers A6c
Convergence acceleration A7
Decomposition of machine-base numbers A6c
Integer A1
Interval A5
Rational A2
Real (single, double, and extended precision, extended range) A3

Sequences **A7**
Type conversion **A6a**

also see MATRICES, VECTORS

ARMA and ARIMA modeling and forecasting (*see* TIME SERIES)
Assignments problem (*see* OPTIMIZATION)
Autocorrelation analysis (*see* TIME SERIES)

B

Balanced designs (*see* EXPERIMENTAL DESIGN)
Barrier method (*see* OPTIMIZATION)
Base conversion (*see* ARITHMETIC)
Bessel functions and their integrals (*see* SPECIAL FUNCTIONS)
Beta and related functions (*see* SPECIAL FUNCTIONS)
Beta distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Biased regression (*see* REGRESSION)
Biharmonic equation (*see* PARTIAL DIFFERENTIAL EQUATIONS)
Binary search (*see* OPTIMIZATION)
Binomial coefficient (*see* SPECIAL FUNCTIONS)
Binomial distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS, STATISTICS)
Bisection method (*see* NONLINEAR EQUATIONS, OPTIMIZATION)
Bit manipulation (*see* DATA HANDLING)
Boolean distribution (*see* RANDOM NUMBERS)
Boundary value problems (*see* ORDINARY DIFFERENTIAL EQUATIONS)
Box plots (*see* GRAPHICS)
Box-Jenkins (*see* TIME SERIES)
Bulirsch-Stoer method (*see* ORDINARY DIFFERENTIAL EQUATIONS)

C

Calibration (inverse regression) (*see* REGRESSION)

CATEGORICAL DATA **L9**

2-by-2 tables **L9a**
EDA (e.g., median polish) **L9d**
Log-linear model **L9c**
Two-way tables **L9b**

Cauchy distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Ceiling (*see* SPECIAL FUNCTIONS)
Character manipulation (*see* DATA HANDLING)
Chebyshev approximation (*see* APPROXIMATION, REGRESSION)
Chebyshev polynomials (*see* SPECIAL FUNCTIONS)
Chebyshev series (*see* SPECIAL FUNCTIONS)
Chi-squared distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Cholesky decomposition (*see* MATRICES)
Clenshaw-Curtis method (*see* QUADRATURE)
Cluster analysis (*see* STATISTICS)
Collocation method (*see* ORDINARY DIFFERENTIAL EQUATIONS, PARTIAL DIFFERENTIAL EQUATIONS)
Complete designs (*see* EXPERIMENTAL DESIGN)
Complex arithmetic (*see* ARITHMETIC)
Complex demodulation (*see* TIME SERIES)

COMPUTATIONAL GEOMETRY (e.g., closest- and farthest-point, line intersection, convex hull, minimum spanning tree, triangulation, Voronoi diagram, polygon intersection, hidden line) **P**

Confidence intervals (*see* STATISTICS)

Confluent hypergeometric functions (*see* SPECIAL FUNCTIONS)

Conjugate gradient method (*see* OPTIMIZATION)

Constrained problems (*see* APPROXIMATION, OPTIMIZATION)

Contingency tables (*see* CATEGORICAL DATA, RANDOM NUMBERS)

Contrasts (*see* EXPERIMENTAL DESIGN)

Convergence acceleration (*see* ARITHMETIC)

Conversion (base and type) (*see* ARITHMETIC)

Convolutions (*see* INTEGRAL TRANSFORMS, VECTORS)

Copy (*see* MATRICES, VECTORS)

Correlation analysis (*see* STATISTICS)

Correlation coefficient (*see* STATISTICS)

Correlation matrix (*see* RANDOM NUMBERS)

Cosine integrals (*see* SPECIAL FUNCTIONS)

Coulomb wave functions (*see* SPECIAL FUNCTIONS)

Covariance, analysis of (*see* EXPERIMENTAL DESIGN)

Covering problems (*see* OPTIMIZATION)

Cross-correlation analysis (*see* TIME SERIES)

Curve fitting (*see* APPROXIMATION, INTERPOLATION, REGRESSION)

Cyclic reduction method (*see* PARTIAL DIFFERENTIAL EQUATIONS)

D

DATA HANDLING **N**

Bit manipulation **N2**

Character manipulation **N3**

Heaps **N4**

Input and output **N1**

Merging **N7**

Permuting **N8**

Ranking **N6a1**

Searching (for extreme value, insertion position, on a key) **N5**

Sorting (internal (passive and active; integer, real, and character) and external) **N6**

Stacks **N4**

Storage management (e.g., stacks, heaps, trees) **N4**

Trees **N4**

Data manipulation (*see* STATISTICS)

Data summarization (*see* STATISTICS)

Dawson's integral (*see* SPECIAL FUNCTIONS)

Decomposition (*see* MATRICES)

Density functions (*see* PROBABILITY FUNCTIONS, STATISTICS)

Determinants (*see* MATRICES)

Diagonalization (*see* MATRICES)

Differential equations (*see* ORDINARY DIFFERENTIAL EQUATIONS, PARTIAL DIFFERENTIAL EQUATIONS)

DIFFERENTIATION **H1**

Splines **H1**

Digital filtering (*see* TIME SERIES)
Discriminant analysis (*see* STATISTICS)
Dispersion (*see* STATISTICS)
Distribution functions (*see* GRAPHICS, PROBABILITY FUNCTIONS, STATISTICS)
Distribution-free analysis (*see* EXPERIMENTAL DESIGN, STATISTICS)
Documentation retrieval (*see* SERVICE ROUTINES)
Domain triangulation (*see* PARTIAL DIFFERENTIAL EQUATIONS)
Dot product (*see* VECTORS)
Double exponential distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Double precision arithmetic (*see* ARITHMETIC)
Dynamic programming (*see* OPTIMIZATION)

E

EDA (exploratory data analysis) (*see* CATEGORICAL DATA, GRAPHICS, REGRESSION)
Eigenvalue problems (*see* MATRICES, PARTIAL DIFFERENTIAL EQUATIONS)
Eigenvalues and eigenvectors (*see* MATRICES)
Elliptic boundary value problems (*see* PARTIAL DIFFERENTIAL EQUATIONS)
Elliptic equations (*see* PARTIAL DIFFERENTIAL EQUATIONS)
Elliptic integrals (*see* SPECIAL FUNCTIONS)
Error analysis (*see* ARITHMETIC)
Error checking (e.g., check monotonicity) (*see* SERVICE ROUTINES)
Error functions, their inverses, and their integrals (*see* PROBABILITY FUNCTIONS, SPECIAL FUNCTIONS)
Error handling (*see* SERVICE ROUTINES)
Errors in variables (*see* REGRESSION)
Euclidean (L_2) norm (*see* MATRICES, VECTORS)
Evaluation of fitted functions (*see* APPROXIMATION, INTERPOLATION)
Exchange (*see* VECTORS)

EXPERIMENTAL DESIGN L7

Analysis of covariance (ANCOVA) L7a3
Analysis of variance (ANOVA) L7a1, L7a2
Balanced complete designs L7a2a
Balanced incomplete designs L7a2b
Contrasts L7a1a1
Distribution-free analysis L7a1b, L7a2a2, L7a2b2, L7a4b
Factorial designs L7a2a1b
General linear model (unbalanced design) L7a4
Latin square L7a2b1a
Lattice designs L7a2b1b
Multiple comparisons L7a1a1
Multivariate analysis of variance (MANOVA) L7b
Nested designs L7a2a1c
Nonparametric analysis L7a1b, L7a2a2, L7a2b2, L7a4b
One-way analysis of variance (ANOVA) L7a1
Repeated measures L7
Student-Newman-Keuls test L7a1a1
Two-way analysis of variance L7a2a1a
Unbalanced designs L7a4
Variance components L7a1a2

Exploratory data analysis (*see* CATEGORICAL DATA, GRAPHICS, REGRESSION)
Exponential distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Exponential functions and integrals (*see* SPECIAL FUNCTIONS)
Extended precision and extended range arithmetic (*see* ARITHMETIC)
Extrapolation methods (*see* ORDINARY DIFFERENTIAL EQUATIONS)

Extreme value distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS, STATISTICS)

F

F-distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)

Factor analysis (*see* STATISTICS)

Factorial (*see* SPECIAL FUNCTIONS)

Factorial designs (*see* EXPERIMENTAL DESIGN)

Fast Fourier transforms (*see* INTEGRAL TRANSFORMS, TIME SERIES)

Filtering (*see* TIME SERIES)

Fletcher and Reeves' method (*see* OPTIMIZATION)

Floor (*see* SPECIAL FUNCTIONS)

Forecasting (*see* TIME SERIES)

Fourier transform (*see* INTEGRAL TRANSFORMS)

Fredholm integral equations (*see* INTEGRAL EQUATIONS)

Fresnel integrals (*see* SPECIAL FUNCTIONS)

Function evaluation (*see* PROBABILITY FUNCTIONS, SPECIAL FUNCTIONS)

G

Galerkin's method (*see* ORDINARY DIFFERENTIAL EQUATIONS, PARTIAL DIFFERENTIAL EQUATIONS)

Gamma and related functions (*see* SPECIAL FUNCTIONS)

Gamma distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)

Gaussian (normal) distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS, STATISTICS)

Gear's method (*see* ORDINARY DIFFERENTIAL EQUATIONS)

General (continuous and discrete) distribution (*see* PROBABILITY FUNCTIONS, RANDOM NUMBERS)

General linear model (*see* EXPERIMENTAL DESIGN)

Generalized inverse (*see* MATRICES)

Geometric distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)

Geometric programming (*see* OPTIMIZATION)

Givens transformation (*see* MATRICES, VECTORS)

Golden section search (*see* OPTIMIZATION)

Goodness-of-fit tests (*see* STATISTICS)

Gram-Schmidt orthogonalization (*see* MATRICES)

GRAPHICS L3, Q

Box plots L3d

Distribution functions L3b

EDA graphics (e.g., stem-and-leaf, box plots) L3d

Histograms L3a

Lag plots L3c5

Line printer plotting Q1, L3

Multiple plots L3c3

Probability plots L3c4 (*specific distributions itemized in subclasses*)

Scatter diagrams L3c

Stem-and-leaf plots L3d

Symbol plots L3c2

Time series plots L3c5

Vertical plots L3c5

Grid generation (*see* INTERPOLATION)

Grouped data (*see* STATISTICS)

H

Halfnormal distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Heaps (*see* DATA HANDLING)
Helmholtz equation (*see* PARTIAL DIFFERENTIAL EQUATIONS)
Hermite polynomials (*see* SPECIAL FUNCTIONS)
Hessenberg matrix (*see* MATRICES)
Hilbert transforms (*see* INTEGRAL TRANSFORMS)
Histograms (*see* GRAPHICS)
Horner's rule (*see* SPECIAL FUNCTIONS)
Householder transformation (*see* MATRICES, VECTORS)
Hyperbolic and inverse hyperbolic functions (*see* SPECIAL FUNCTIONS)
Hyperbolic initial boundary value problems (*see* PARTIAL DIFFERENTIAL EQUATIONS)
Hypergeometric distribution (*see* PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Hypergeometric function (*see* SPECIAL FUNCTIONS)
Hypothesis testing (*see* STATISTICS)

I

Incomplete designs (*see* EXPERIMENTAL DESIGN)
Inference (*see* STATISTICS)
Initial boundary value problems (*see* PARTIAL DIFFERENTIAL EQUATIONS)
Initial value problems (*see* ORDINARY DIFFERENTIAL EQUATIONS)
Inner product (*see* VECTORS)
Input (*see* DATA HANDLING)
Integer arithmetic (*see* ARITHMETIC)
Integer programming (*see* OPTIMIZATION)
Integers (*see* RANDOM NUMBERS)

INTEGRAL EQUATIONS I3

Fredholm integral equations I3
Volterra integral equations I3

INTEGRAL TRANSFORMS J

Convolutions J2
Fast Fourier transforms J1
Hilbert transforms J4
Laplace transforms J3
Multidimensional fast Fourier transforms J1b
One-dimensional fast Fourier transforms (real, complex, and trigonometric) J1

also see QUADRATURE, TIME SERIES

Integrals (*see* SPECIAL FUNCTIONS)
Integration (*see* QUADRATURE, ORDINARY DIFFERENTIAL EQUATIONS)

INTERPOLATION E

Aitken's method E1b
Curve fitting E1
Evaluation of fitted functions E3
Grid generation E3
Multivariate (for gridded and for scattered data) E2

Piecewise polynomials **E1a**
Polynomial splines **E1a**
Polynomials **E1b**
Rational functions **E1c**
Service routines (e.g., grid generation, evaluation of fitted functions) **E3**
Splines **E1a, E2a, E3**
Surface fitting **E2**
Trigonometric functions **E1c**
Univariate **E1**

Interval arithmetic (*see* ARITHMETIC)
Inverse distribution functions (*see* PROBABILITY FUNCTIONS)
Inverse regression (*see* REGRESSION)
Inversion (*see* LINEAR EQUATIONS, MATRICES)

J

Jacobi method (*see* LINEAR EQUATIONS, MATRICES, PARTIAL DIFFERENTIAL EQUATIONS)
Jacobian elliptic functions (*see* SPECIAL FUNCTIONS)
Jordan normal form (*see* MATRICES)

K

Kalman filtering (*see* TIME SERIES)
Kelvin functions (*see* SPECIAL FUNCTIONS)
Kendall's coefficient of concordance (*see* STATISTICS)
Kendall's rank correlation coefficient (*see* STATISTICS)
Knapsack problems (*see* OPTIMIZATION)
Kolmogorov-Smirnov distribution (*see* PROBABILITY FUNCTIONS)
Kolmogorov-Smirnov test (*see* STATISTICS)
Kurtosis (*see* STATISTICS)

L

L_1 (*see* APPROXIMATION, REGRESSION, VECTORS)
 L_2 (*see* APPROXIMATION, REGRESSION, VECTORS)
 L_∞ (*see* APPROXIMATION, REGRESSION, VECTORS)
Lack-of-fit tests (*see* REGRESSION)
Lag plots (*see* GRAPHICS)
Laguerre method (*see* NONLINEAR EQUATIONS)
Laguerre polynomials (*see* SPECIAL FUNCTIONS)
Laguerre quadrature (*see* QUADRATURE)
Lambda distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Laplace equation (*see* PARTIAL DIFFERENTIAL EQUATIONS)
Laplace transforms (*see* INTEGRAL TRANSFORMS)
Latin square (*see* EXPERIMENTAL DESIGN)
Lattice designs (*see* EXPERIMENTAL DESIGN)
Leaps-and-bounds algorithm (*see* REGRESSION)
Least absolute value (L_1) (*see* APPROXIMATION, REGRESSION)
Least squares (L_2) (*see* APPROXIMATION, REGRESSION)
Legendre functions (*see* SPECIAL FUNCTIONS)
Legendre polynomials (*see* SPECIAL FUNCTIONS)
Levenberg-Marquardt method (*see* APPROXIMATION, REGRESSION)
Life testing (*see* STATISTICS)

Line-printer plotting (*see* GRAPHICS)

Linear algebra (*see* MATRICES, VECTORS)

LINEAR EQUATIONS D2, D8, D9

- Associated operations (e.g., matrix reorderings) D2e
- Banded systems D2a2, D2b2, D2c2, D2d2
- Complex Hermitian systems D2d
- Complex non-Hermitian systems D2c
- Hermitian indefinite systems D2d1a
- Jacobi method D2a4, D2b4
- Other matrix equations (e.g., $AX + BX = C$) D8
- Overdetermined systems of equations D9
- Positive definite systems D2b1b, D2b2, D2d1b, D2d2
- Real nonsymmetric systems D2a
- Real symmetric systems D2b
- Relaxation methods D2b4
- Singular systems D9
- SOR method D2b4
- Sparse systems D2a4, D2b4, D2c4, D2d4
- Successive over-relaxation method D2b4
- Symmetric indefinite systems D2b1a
- Triangular systems D2a3, D2c3
- Tridiagonal systems D2a2a, D2b2a, D2c2a, D2d2a
- Underdetermined systems of equations D9

also see MATRICES

Linear programming (*see* OPTIMIZATION)

Location (*see* STATISTICS)

Location problems (*see* OPTIMIZATION)

Log gamma function (*see* SPECIAL FUNCTIONS)

Log-linear model (*see* CATEGORICAL DATA)

Logarithmic functions and integrals (*see* SPECIAL FUNCTIONS)

Logical distribution (*see* RANDOM NUMBERS)

Logistic distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)

Logistic regression (*see* REGRESSION)

Lognormal distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)

LR algorithm (*see* MATRICES)

LU decomposition (*see* MATRICES)

M

Machine-dependent constants (*see* SERVICE ROUTINES)

Mann-Whitney rank-sum test (*see* STATISTICS)

Markov models (*see* SIMULATION AND STOCHASTIC MODELING)

Matching problems (*see* OPTIMIZATION)

Mathieu functions (*see* SPECIAL FUNCTIONS)

MATRICES D

- Addition and subtraction D1b5
- Bandwidth and profile reduction D2e
- Cholesky decomposition D2b1b
- Copy D1b8
- Determinants D3

Diagonalization **D4**

Eigenvalues and eigenvectors (ordinary eigenvalue problems, generalized eigenvalue problems and associated operations (e.g., transform problem, balance matrix, reduce to compact form, standardize problem, compute eigenvalues of matrix in compact form, form eigenvectors from eigenvalues, back transform eigenvectors) **D4**

Euclidean (L_2) norm **D1b2**

Generalized inverse **D9**

Givens transformation **D1b10**

Gram-Schmidt orthogonalization **D5**

Hessenberg matrix **D4**

Householder transformation **D1b11**

Inversion **D2**

Jacobi method **D4a1, D4b1**

Jordan normal form **D4c5**

LR algorithm **D4c**

LU decomposition **D2, D7a**

Matrix polynomial **D1b7**

Multiplication **D1b6**

Multiplication by vector **D1b4**

Norm **D1b2**

Pseudo-inverse **D9**

QR decomposition **D5**

QZ algorithm **D4**

Reflection **D1b11**

Reorderings (of rows and columns) **D2e**

Rotation **D1b10**

Set to zero and identity **D1b1**

Singular value decomposition **D6**

Storage mode conversion **D1b9**

Transpose **D1b3**

Update matrix decompositions (LU, Cholesky, QR, and singular value) **D7**

also see LINEAR EQUATIONS

Maximum flow problems (*see* OPTIMIZATION)

Maximum likelihood (*see* STATISTICS)

Median polish (*see* CATEGORICAL DATA)

Merging (*see* DATA HANDLING)

Mesh generation (*see* APPROXIMATION)

Minimax (L_∞) (*see* APPROXIMATION, REGRESSION)

Minimum spanning tree (*see* OPTIMIZATION)

Missing values (*see* REGRESSION, STATISTICS)

Moments (*see* STATISTICS)

Monte Carlo integration (*see* QUADRATURE)

Moving averages (*see* TIME SERIES)

Muller's method (*see* NONLINEAR EQUATIONS)

Multi-dimensional integrals (*see* QUADRATURE)

Multinomial distribution (*see* RANDOM NUMBERS)

Multiphase regression (*see* REGRESSION)

Multiple comparisons (*see* EXPERIMENTAL DESIGN)

Multiple regression (*see* REGRESSION)

Multivariate analysis of variance (*see* EXPERIMENTAL DESIGN)

Multivariate distribution and density functions (*see* PROBABILITY FUNCTIONS)

Multivariate regression (*see* REGRESSION)

N

Negative binomial distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)

Nested designs (*see* EXPERIMENTAL DESIGN)

Network reliability (*see* SIMULATION AND STOCHASTIC MODELING)

Network problems (*see* OPTIMIZATION)

NONLINEAR EQUATIONS F

Bisection method F1b

Laguerre method F1a1a

Muller's method F1a2

Polynomial (real and complex coefficients) F1a1

Service routines (e.g., check user-supplied derivatives) F3

Single nonlinear equation (smooth or general (no smoothness assumed)) F1

System of equations (smooth or general (no smoothness assumed)) F2

Nonlinear problems (*see* APPROXIMATION, NONLINEAR EQUATIONS, OPTIMIZATION, REGRESSION)

Nonparametric statistics (*see* EXPERIMENTAL DESIGN, STATISTICS)

Nonseparable elliptic boundary value problems (*see* PARTIAL DIFFERENTIAL EQUATIONS)

Norm (*see* MATRICES, VECTORS)

Normal distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS, STATISTICS)

Normal scores (*see* PROBABILITY FUNCTIONS)

NUMBER THEORY (e.g., prime numbers) B

O

One-way analysis of variance (*see* EXPERIMENTAL DESIGN)

Optimal control (*see* OPTIMIZATION)

OPTIMIZATION G

Assignments problem G2b

Barrier method G2h3

Binary search G1a2

Bisection method G1a2

Conjugate gradient method G1b1

Constrained G2

Covering problems G2c2

Dynamic programming G2g

Fletcher and Reeves' method G1b1

Geometric programming G2f

Global solution to nonconvex problems G2i

Golden section search G1a2

Integer programming G2c

Knapsack problems G2c3

Linear programming (including simplex method) G2a

Location problems G2c5

Matching problems G2c4

Maximum flow problems G2d3

Minimum spanning tree G2d2

Network problems (including test problem generation) G2d

Nonlinear programming (simple bounds, linear equality or inequality constraints, nonlinear constraints) G2h

Optimal control G3

Packing problems G2c2

Penalty method G2h

Quadratic programming **G2e**
Quasi-Newton methods **G1b1**
Routing problems **G2c5**
Scheduling problems **G2c5**
Service routines (including problem input (e.g., matrix generation), problem scaling, checking
user-supplied derivatives, finding feasible point, and checking for redundancy) **G4**
Shortest path problems **G2d1**
Simplex method **G1b2, G2a**
Transportation problems **G2b**
Unconstrained **G1**

Order statistics (*see* RANDOM NUMBERS)

ORDINARY DIFFERENTIAL EQUATIONS **I1**

Adams' predictor-corrector method **I1a1b**
Algebraic-differential systems **I1a2**
Boundary value problems **I1b**
Bulirsch-Stoer method **I1a1c**
Collocation method **I1b**
Extrapolation methods **I1a1c**
Galerkin's method **I1b**
Gear's method **I1a2**
Initial value problems **I1a**
Initial value problems - general, non-stiff or mildly stiff **I1a1**
Initial value problems - stiff and mixed algebraic-differential equations **I1a2**
Multi-point boundary value problems (linear, nonlinear, and eigenvalue problems) **I1b**
Multistep methods **I1a1b**
One-step methods **I1a1a**
Predictor-corrector methods **I1a1b**
Runge-Kutta methods **I1a1a**
Service routines (interpolation of solutions, error handling) **I1c**
Shooting method **I1b**
Stiff initial value problems **I1a2**
Sturm-Liouville **I1b3**

Ordinary least squares (*see* REGRESSION)
Oscillatory integrals (*see* QUADRATURE)
Orthogonal matrix (*see* RANDOM NUMBERS)
Orthogonal polynomials (*see* REGRESSION, SPECIAL FUNCTIONS)
Output (*see* DATA HANDLING)
Overdetermined systems of equations (*see* LINEAR EQUATIONS)

P

Packing problems (*see* OPTIMIZATION)
Pade approximation (*see* APPROXIMATION)
Parabolic cylinder functions (*see* SPECIAL FUNCTIONS)
Parabolic initial boundary value problems (*see* PARTIAL DIFFERENTIAL EQUATIONS)
Parameter estimation (*see* STATISTICS)
Pareto distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)

PARTIAL DIFFERENTIAL EQUATIONS **I2**

Biharmonic equation **I2b1c**
Collocation method **I2**

Cyclic reduction method **I2b4b**
 Domain triangulation **I2b4a**
 Elliptic boundary value problems (linear, nonlinear, and eigenvalue problems) **I2b**
 Galerkin's method **I2**
 Helmholtz equation **I2b1a1**
 Hyperbolic initial boundary value problems **I2a2**
 Initial boundary value problems **I2a**
 Jacobi method **I2b4b**
 Laplace equation **I2b1a1**
 Nonseparable elliptic boundary value problems **I2b1a3**
 Parabolic initial boundary value problems **I2a1**
 Poisson equation **I2b1a1**
 Relaxation methods **I2b4b**
 Separable elliptic boundary value problems **I2b1a1, I2b1a2**
 Service routines (including domain triangulation and solution of discretized elliptic equations) **I2b4**
 SOR method **I2b4b**
 Stone's procedure **I2b4b**
 Strongly implicit procedure **I2b4b**
 Successive over-relaxation method **I2b4b**
 Triangulation **I2b4a**

Penalty method (*see* OPTIMIZATION)
 Permutations (*see* DATA HANDLING, RANDOM NUMBERS)
 PERT (*see* SIMULATION AND STOCHASTIC MODELING)
 Piecewise polynomials (*see* APPROXIMATION, INTERPOLATION, REGRESSION)
 Plotting (*see* GRAPHICS)
 Poisson distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS, STATISTICS)
 Poisson equation (*see* PARTIAL DIFFERENTIAL EQUATIONS)
 Polygamma function (*see* SPECIAL FUNCTIONS)
 Polynomial splines (*see* APPROXIMATION, INTERPOLATION, REGRESSION)
 Polynomials (*see* APPROXIMATION, INTERPOLATION, NONLINEAR EQUATIONS, REGRESSION, SPECIAL FUNCTIONS)
 Powers (*see* SPECIAL FUNCTIONS)
 Predictor-corrector methods (*see* ORDINARY DIFFERENTIAL EQUATIONS)
 Prime numbers (*see* NUMBER THEORY)
 Principal components (*see* REGRESSION, STATISTICS)
 Principal value integrals (*see* QUADRATURE)
 Probability (*see* STATISTICS)

PROBABILITY FUNCTIONS **L5**

Density and distribution functions **L5a1** (*specific distributions itemized in subclasses*)
 Inverse distribution functions (percent point functions) **L5a2** (*specific distributions itemized in subclasses*)
 Multivariate distribution and density functions **L5b1** (*specific distributions itemized in subclasses*)
 Sparsity functions **L5a2** (*specific distributions itemized in subclasses*)

also see SPECIAL FUNCTIONS

Probability plots (*see* GRAPHICS)
 Project optimization (e.g., PERT) (*see* SIMULATION AND STOCHASTIC MODELING)
 Proportional data (*see* STATISTICS)
 Pseudo-inverse (*see* MATRICES)
 Psi function (*see* SPECIAL FUNCTIONS)

Q

QR decomposition (*see* MATRICES)

Quadratic programming (*see* OPTIMIZATION)

QUADRATURE (numerical evaluation of definite integrals) **H2**

Adaptive quadrature **H2**

Clenshaw-Curtis method **H2a2a**

Finite interval integrals (general integrand) **H2a1**

Finite interval integrals (special integrands including weight functions, oscillatory and singular integrands, principal value integrals, splines) **H2a2**

Infinite interval integrals (including $\exp(-x^2)$ weight function) **H2a4**

Laguerre quadrature **H2c**

Monte Carlo integration **H2b1a2**

Multi-dimensional integrals over hyper-rectangular regions **H2b1**

Multi-dimensional integrals over nonrectangular regions **H2b2**

Oscillatory integrals **H2a2**

Principal value integrals **H2a2**

Romberg integration **H2a1a1**

Semi-infinite interval integrals (including $\exp(-x)$ weight function) **H2a3**

Service routines (compute weights and nodes for quadrature formulas) **H2c**

Splines **H2a1b2, H2a2a1**

Qualitative data (*see* STATISTICS)

Quality control (*see* SIMULATION AND STOCHASTIC MODELING)

Quasi-Newton methods (*see* OPTIMIZATION)

Queueing (*see* SIMULATION AND STOCHASTIC MODELING)

QZ algorithm (*see* MATRICES)

R

RANDOM NUMBERS **L6**

Multivariate **L6b** (*specific distributions itemized in subclasses*)

Univariate **L6a** (*specific distributions itemized in subclasses*)

Rank tests (*see* STATISTICS)

Ranking (*see* DATA HANDLING)

Rational arithmetic (*see* ARITHMETIC)

Rational functions (*see* APPROXIMATION, INTERPOLATION)

Real arithmetic (*see* ARITHMETIC)

Reciprocal gamma function (*see* SPECIAL FUNCTIONS)

Reciprocals (*see* SPECIAL FUNCTIONS)

Rectangular (uniform) distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS, STATISTICS)

Reflection (*see* MATRICES, VECTORS)

REGRESSION **L8**

Biased (ridge) **L8b**

Calibration (inverse regression) **L8a1d**

Chebyshev **L8d**

Correlation data, from **L8a4a1c**

Design **L8a6**

Diagnostics **L8a9**

EDA (exploratory data analysis) **L8f**

Errors in variables **L8a1c, L8a4b**

Exploratory data analysis **L8f**
Hypothesis testing **L8a10**
Inference **L8a10**
Inverse regression **L8a1d**
 L_1 **L8c**
 L_2 **L8a**
 L_∞ **L8d**
Lack-of-fit tests **L8a10a**
Leaps-and-bounds algorithm **L8a5**
Levenberg-Marquardt method **L8g1a, L8g2a**
Linear least absolute value (L_1) **L8c**
Linear least squares (L_2) **L8a**
Linear minimax (L_∞ or Chebyshev) **L8d**
Logistic **L8a4d**
Missing values **L8a1a1b, L8a4a1b**
Multiphase **L8a3**
Multiple **L8a4**
Multivariate **L8a8**
Nonlinear **L8g**
Ordinary **L8a1a, L8a4a**
Orthogonal polynomials **L8a2b**
Piecewise polynomial **L8a3**
Polynomial **L8a2**
Polynomial spline **L8a3**
Preference pairs, using **L8a4a1e**
Principal components, using **L8a4a1d**
Residual analysis **L8a10b**
Ridge **L8b**
Robust **L8e**
Several multiple regressions **L8a7**
Simple **L8a1**
Spline **L8a3**
Stepwise **L8a5**
Through the origin **L8a1b**
Unweighted **L8a1a1, L8a2a1, L8a2b1, L8a4a1, L8g1**
Variable selection **L8a5**
Weighted **L8a1a2, L8a2a2, L8a2b2, L8a4a2, L8g2**

also see APPROXIMATION

Regression design (*see* REGRESSION)
Relaxation methods (*see* LINEAR EQUATIONS, PARTIAL DIFFERENTIAL EQUATIONS)
Reliability (*see* SIMULATION AND STOCHASTIC MODELING)
Repeated measures (*see* EXPERIMENTAL DESIGN)
Residual analysis (*see* REGRESSION)
Ridge regression (*see* REGRESSION)
Riemann zeta function (*see* SPECIAL FUNCTIONS)
Robust regression (*see* REGRESSION)
Romberg integration (*see* QUADRATURE)
Roots (*see* SPECIAL FUNCTIONS)
Rotation (*see* MATRICES, VECTORS)
Routing problems (*see* OPTIMIZATION)
Runge-Kutta methods (*see* ORDINARY DIFFERENTIAL EQUATIONS)

S

Samples (*see* RANDOM NUMBERS)
Scale (*see* STATISTICS)
Scatter diagrams (*see* GRAPHICS)
Scheduling problems (*see* OPTIMIZATION)
Scorer functions (*see* SPECIAL FUNCTIONS)
Searching (*see* DATA HANDLING)
Separable elliptic boundary value problems (*see* PARTIAL DIFFERENTIAL EQUATIONS)
Sequences (*see* ARITHMETIC, STATISTICS)

SERVICE ROUTINES R

Documentation retrieval R4
Error checking (e.g., check monotonicity) R2
Error handling (including setting criteria for fatal error, setting unit number for error messages) R3
Machine-dependent constants R1

Shape (*see* STATISTICS)
Shooting method (*see* ORDINARY DIFFERENTIAL EQUATIONS)
Shortest path problems (*see* OPTIMIZATION)
Simple regression (*see* REGRESSION)
Simplex method (*see* OPTIMIZATION)

SIMULATION AND STOCHASTIC MODELING M

Project optimization (e.g., PERT) M4
Queueing M2
Reliability (quality control and electrical network) M3
Simulation (discrete and continuous (Markov models)) M1

Sine integrals (*see* SPECIAL FUNCTIONS)
Single precision arithmetic (*see* ARITHMETIC)
Singular value decomposition (*see* MATRICES)
Skewness (*see* STATISTICS)
Smoothing (*see* APPROXIMATION, TIME SERIES)

SOFTWARE DEVELOPMENT TOOLS S

Dynamic analysis S3
Program transformation S1
Static analysis S2

SOR method (*see* LINEAR EQUATIONS, PARTIAL DIFFERENTIAL EQUATIONS)
Sorting (*see* DATA HANDLING)
Sparsity functions (*see* PROBABILITY FUNCTIONS)
Spearman rank-correlation coefficient (*see* STATISTICS)

SPECIAL FUNCTIONS C

Airy function C10d
Anger functions C10e
Bessel functions C10
Bessel functions, integrals of C10f
Beta and related functions C7b, C7f
Binomial coefficient C1
Ceiling C1
Chebyshev polynomials C3a2
Chebyshev series C3a2

Confluent hypergeometric functions C11
Cosine integrals C6
Coulomb wave functions C12
Dawson's integral C8c
Elliptic integrals C14
Error functions, their inverses, and their integrals C8
Exponential functions C4b
Exponential integrals C5
Factorial C1
Floor C1
Fresnel integrals C8b
Gamma and related functions C7
Hermite polynomials C3a4
Horner's rule C3
Hyperbolic and inverse hyperbolic functions C4c
Hypergeometric function C11
Integer-valued functions (including floor, ceiling, factorial, and binomial coefficient) C1
Integrals of elementary transcendental functions C4d
Jacobian elliptic functions C13
Kelvin functions C10c
Laguerre polynomials C3a3
Legendre functions C9
Legendre polynomials C3a2
Logarithmic functions C4b
Logarithmic integrals C5
Mathieu functions C17
Non-orthogonal polynomials C3b
Orthogonal polynomials C3a
Parabolic cylinder functions C16
Polygamma function C7d
Polynomials C3
Powers C2
Psi function C7c
Reciprocal gamma function C7a
Reciprocals C2
Riemann zeta function C7g
Roots C2
Scorer functions C10d
Sine integrals C6
Spheroidal wave functions C18
Struve functions C10e
Theta functions C13
Transcendental functions C4
Trigonometric and inverse trigonometric functions C4a
Trigonometric integrals C6
Trigonometric polynomials C3a1
Weber functions C10e
Weierstrass elliptic functions C15

also see PROBABILITY FUNCTIONS

Spectral analysis (*see* TIME SERIES)

Spheroidal wave functions (*see* SPECIAL FUNCTIONS)

Splines (*see* APPROXIMATION, DIFFERENTIATION, INTERPOLATION, QUADRATURE, REGRESSION)

Stable distribution (*see* RANDOM NUMBERS)

Stacks (*see* DATA HANDLING)

Standard deviation (*see* STATISTICS)

STATISTICS L

Analysis of variance L7
Categorical data analysis L9
Cluster analysis L14
Confidence intervals L4
Contingency tables L2b, L6b3, L9
Correlation analysis L10c, L10g, L11
Correlation coefficient L1e1b, L4b, L4d
Density functions L1a1d, L4a1c, L5a1, L5b1
Discriminant analysis L12
Dispersion L1a1b, L1a3b, L4
Distribution functions L1a1d, L4a1c, L5a1, L5b1
Distribution-free analysis L4a1b, L4b1b, L4e1b, L4e2b, L4e3b, L7a1b, L7a2a2, L7a2b2, L7a4b
EDA (experimental data analysis) L3d, L8f, L9d
Experimental design L7
Factor analysis L13
Goodness-of-fit tests L4a1c
Graphics L3
Grouped data L1a3, L1e3, L2b, L4a3, L4b3, L4e3
Hypothesis testing L4
Inference L4
Kendall's coefficient of concordance L4a1b
Kendall's rank correlation coefficient L4b1b
Kolmogorov-Smirnov test L4a1c, L4b1b
Kurtosis L1a1c, L1a3c
Life testing L15
Location L1a1a, L1a3a, L1e1a, L4
Manipulation, data L2
Mann-Whitney rank-sum test L4a1b
Maximum likelihood L7a4, L8a4d, L8g, L10e, L10g1, L13
Missing values L1a2, L1e2, L4a2, L4b2, L4e2, L8a1a1b, L8a4a1b
Moments L1, L4
Nonparametric statistics L4a1b, L4b1b, L4e1b, L4e2b, L4e3b, L7a1b, L7a2a2, L7a2b2, L7a4b
Parameter estimation (e.g., binomial, extreme value, normal or Gaussian, Poisson, uniform or rectangular, Weibull) L4a
Principal components L8a4a1d, L13a
Probability functions L5
Proportional data L1c, L1e, L4c, L4d, L4e
Qualitative data L1c, L1e, L4c, L4d, L4e
Random numbers L6
Rank tests L4a1b, L4b1b
Regression L8
Sample L2c
Scale L1a1b, L1a3b, L4
Sequences of numbers L4a1d
Shape L1a1c, L1a3c
Skewness L1a1c, L1a3c
Spearman rank-correlation coefficient L4a1b
Standard deviation L1a1b, L1a3b, L4
Subset L2d
Summarization, data L1
Survival analysis L15
Time series L10
Tolerance limits L4a1f
Transformations L2a, L10a

Variance **L1a1b, L1a3b, L4**
Wilcoxon rank-sum test **L4b1b**
Wilcoxon signed-rank test **L4a1b**

Stem-and-leaf (*see* GRAPHICS)
Stiff initial value problems (*see* ORDINARY DIFFERENTIAL EQUATIONS)
Stone's procedure (*see* PARTIAL DIFFERENTIAL EQUATIONS)
Storage management (*see* DATA HANDLING)
Storage mode conversion (*see* MATRICES)
Strongly implicit procedure (*see* PARTIAL DIFFERENTIAL EQUATIONS)
Struve functions (*see* SPECIAL FUNCTIONS)
Sturm-Liouville (*see* ORDINARY DIFFERENTIAL EQUATIONS)
Successive over-relaxation method (*see* LINEAR EQUATIONS, PARTIAL DIFFERENTIAL EQUATIONS)
Surface fitting (*see* APPROXIMATION, INTERPOLATION)
Survival analysis (*see* STATISTICS)
Swap (*see* VECTORS)
Symbol plots (*see* GRAPHICS)

SYMBOLIC COMPUTATION **O**

System of equations (*see* NONLINEAR EQUATIONS)

T

t-distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Taylor polynomial approximation (*see* APPROXIMATION)
Theta functions (*see* SPECIAL FUNCTIONS)

TIME SERIES **L10**

ARMA and ARIMA modeling and forecasting **L10e**
Autocorrelation analysis **L10c**
Box-Jenkins analysis **L10e**
Complex demodulation **L10d**
Cross-correlation analysis **L10g**
Filtering **L10b**
Kalman filtering **L10b**
Moving averages **L10b, L10e1**
Plots (*see* GRAPHICS)
Random number generation **L6a20**
Smoothing **L10b**
Spectral analysis **L10f**
Transformations **L10a**
Transforms **L10a** (*also see* INTEGRAL EQUATIONS)

Tolerance limits (*see* STATISTICS)
Transcendental functions and their integrals (*see* SPECIAL FUNCTIONS)
Transformations (*see* MATRICES, STATISTICS, TIME SERIES, VECTORS)
Transforms (*see* INTEGRAL TRANSFORMS, TIME SERIES)
Transportation problems (*see* OPTIMIZATION)
Transpose (*see* MATRICES)
Trees (*see* DATA HANDLING)
Triad (*see* VECTORS)
Triangular distribution (*see* RANDOM NUMBERS)
Triangulation (*see* PARTIAL DIFFERENTIAL EQUATIONS)
Trigonometric and inverse trigonometric functions (*see* SPECIAL FUNCTIONS)

Trigonometric functions (*see* APPROXIMATION, INTERPOLATION)
 Trigonometric integrals (*see* SPECIAL FUNCTIONS)
 Trigonometric polynomials (*see* SPECIAL FUNCTIONS)
 Two-way analysis of variance (*see* EXPERIMENTAL DESIGN)
 Two-way tables (*see* CATEGORICAL DATA)
 Type conversion (*see* ARITHMETIC)

U

Unbalanced design (*see* EXPERIMENTAL DESIGN)
 Unconstrained approximation (*see* APPROXIMATION)
 Unconstrained optimization (*see* OPTIMIZATION)
 Underdetermined systems of equations (*see* LINEAR EQUATIONS)
 Uniform distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS, STATISTICS)

V

Variable selection in regression (*see* REGRESSION)
 Variance (*see* STATISTICS)
 Variance components (*see* EXPERIMENTAL DESIGN)

VECTORS D1a

Convolutions **D1a10**
 Copy **D1a5**
 Dot product **D1a4**
 Exchange **D1a5**
 Euclidean (L_2) norm **D1a3b**
 Givens transformation **D1a8**
 Householder transformation **D1a9**
 Inner product **D1a4**
 L_1 norm (sum of magnitudes) **D1a3a**
 L_2 norm **D1a3b**
 L_∞ norm **D1a3c**
 Minimum and maximum components **D1a2**
 Multiplication by scalar **D1a6**
 Norm **D1a3**
 Reflection **D1a9**
 Rotation **D1a8**
 Set to constant **D1a1**
 Swap **D1a5**
 Triad ($ax+by$ for vectors x, y and scalar a) **D1a7**

Vertical plots (*see* GRAPHICS)
 Volterra integral equation (*see* INTEGRAL EQUATIONS)
 Von Mises distribution (*see* RANDOM NUMBERS)

W

Weber functions (*see* SPECIAL FUNCTIONS)
 Weibull distribution (*see* GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS, STATISTICS)
 Weierstrass elliptic functions (*see* SPECIAL FUNCTIONS)
 Weighted regression (*see* REGRESSION)
 Wilcoxon rank-sum test (*see* STATISTICS)
 Wilcoxon signed-rank test (*see* STATISTICS)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET <i>(See instructions)</i>	1. PUBLICATION OR REPORT NO. NBSIR 84-2824	2. Performing Organ. Report No.	3. Publication Date February 1984
4. TITLE AND SUBTITLE Guide to Available Mathematical Software (GAMS)			
5. AUTHOR(S) Ronald F. Boisvert, Sally E. Howe, David K. Kahaner			
6. PERFORMING ORGANIZATION <i>(If joint or other than NBS, see instructions)</i> NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234		7. Contract/Grant No. 8. Type of Report & Period Covered	
9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS <i>(Street, City, State, ZIP)</i> 			
10. SUPPLEMENTARY NOTES <input type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.			
11. ABSTRACT <i>(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)</i> The second edition of the Guide to Available Mathematical Software (GAMS) provides summary documentation of software available to NBS staff on a variety of computers. The fifteen libraries documented in GAMS are: BMDP, CMLIB, (containing three dozen public-domain packages), DATAPAC, IMSL, INVAR, MATHWARE, MATLAB, MINITAB, NAG, PDELIB, PLOD, PORT, SLDGL, SPECTRLAN, STATLIB. GAMS is based on an extensive problem-oriented scheme for classifying software for mathematical computations including special functions, linear algebra, optimization, differentiation and integration, differential and integral equations, and statistics and probability. The document contains the classification scheme, a catalog of software organized by class, a dictionary of the software, library references, and an index.			
12. KEY WORDS <i>(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)</i> Catalog, classification scheme, documentation, mathematical software, statistical software, software			
13. AVAILABILITY <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input type="checkbox"/> Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. <input checked="" type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161			14. NO. OF PRINTED PAGES 448 15. Price \$32.50

GAMS HELP

The following people can provide assistance using GAMS. This includes identifying the most appropriate software, using software, or finding further information. Please contact one of them if you find errors in this document.

Help using software on the Sperry 1100/82 and Perkin-Elmer minicomputers in Gaithersburg is available from:

Paul Boggs	numerical analysis	(301 or FTS 921-3395)
Ronald Boisvert	numerical analysis	(301 or FTS 921-3395)
Elsie Clark	statistics	(301 or FTS 921-3395)
Karla Hoffman	linear and nonlinear programming	(301 or FTS 921-3855)
Sally Howe	statistics	(301 or FTS 921-3395)
Ric Jackson	optimization	(301 or FTS 921-3855)
David Kahaner	numerical analysis	(301 or FTS 921-3395)
Janice Knapp-Cordes	IMSL, NAG, and PORT	(301 or FTS 921-3395)
Daniel Lozier	special functions	(301 or FTS 921-2631)
Patsy Saunders	simulation	(301 or FTS 921-3855)

To establish an account on the Sperry 1100, or for other assistance in using the central computing facility in Gaithersburg, please contact the NBS Computer Services Division at 301 or FTS 921-3580 (for accounts, call 301 or FTS 921-3364).

Help using software on the Cyber 750 and Perkin-Elmer minicomputers in Boulder is available from:

Janet Donaldson	statistics	(303-497-5114 or FTS 320-5114)
Lloyd Fosdick	numerical analysis	(303 497-3836 or FTS 320-3836)
John Gary	numerical analysis	(303 497-3369 or FTS 320-3369)
John Koontz	statistics	(303 497-5180 or FTS 320-5180)
Linda Lindgren	numerical analysis	(303 497-5149 or FTS 320-5149)
Roland Sweet	numerical analysis	(303 497-5671 or FTS 320-5671)

To establish an account on the Cyber 750, or for other assistance in using the central computing facility in Boulder, please contact the NOAA Computer Services Division at (303) 497-5850 or FTS 320-5850.

